Error estimation and adaptive mesh refinement for aerodynamic flows

Ralf Hartmann[1] and Paul Houston[2]

[1] *Institute of Aerodynamics and Flow Technology*
DLR (German Aerospace Center)
Lilienthalplatz 7, 38108 Braunschweig, Germany
Ralf.Hartmann@dlr.de

[2] *School of Mathematical Sciences*
University of Nottingham
University Park, Nottingham, NG7 2RD, UK
Paul.Houston@nottingham.ac.uk

Contents

1 Introduction

Computational fluid dynamics (CFD) has become a key technology in the development of new products in the aeronautical industry. During the last decade aerodynamic design engineers have progressively adapted their way-of-working to take advantage of the possibilities offered by new CFD capabilities based on the solution of the Euler and Navier–Stokes equations. Significant improvements in physical modelling and solution algorithms have been as important as the enormous increase of computer power to enable numerical simulations at all stages of aircraft development.

However, despite the progress made in CFD, in terms of user time and computational resources, large aerodynamic simulations of viscous flows around complex configurations are still very expensive. The requirement to reliably compute results with a sufficient level of accuracy within short turn-around times places severe constraints on the application of CFD. Indeed, within CFD the most popular class of methods which are currently used in industrial codes are based on employing finite volume methods. While in principal these methods are second–order accurate, in practice their convergence order deteriorates to somewhere between first– and second–order on irregular and/or highly stretched meshes. Thereby, for reliable numerical predictions to be made by such methods, extremely fine meshes with a large number of degrees of freedom are required, which in turn leads to excessively large computing times. As an alternative approach, in recent years there has been significant interest in the development of high–order discretization methods; this is particularly evidenced by the funding of the EU Framework 6 project ADIGMA [82] (Adaptive higher order variational methods for aerospace applications) comprising of a consortium of academic and industrial partners. On a given mesh they allow for an improved prediction of critical flow phenomena, such as boundary layers, wakes, and vortices, for example, as well as force coefficients, e.g., drag, lift, moment. In particular, high–order methods are capable of achieving the same level of accuracy while exploiting significantly fewer degrees of freedom compared with classical finite volume methods.

One extremely promising class of high-order schemes based on the finite element framework are Discontinuous Galerkin (DG, for short) methods. Indeed, the development of DG methods for the numerical approximation of the Euler and Navier-Stokes equations is an extremely exciting research topic which is currently being developed by a number of groups all over the world, cf. [14, 15, 19, 20, 34, 38, 39, 50, 59, 61, 62, 95, 107, 108], for example. DG methods have several important advantages over well established finite volume methods. The concept of higher-order discretization is inherent to the DG method. The stencil is minimal in the sense that each element communicates only with its direct neighbors. In particular, in contrast to the increasing stencil size needed to increase the accuracy of classical finite volume methods, the stencil of DG methods is the same for any order of accuracy which has important advantages for the implementation of boundary conditions and for the parallel efficiency of the method. Moreover, due this simple communication at element interfaces, elements with so–called hanging nodes can be easily treated, a fact that simplifies local mesh refinement (h–refinement). Additionally, the communication at element interfaces is identical for any order of the method which simplifies the use of methods with different polynomial orders p in adjacent elements. This allows for the variation of the order of polynomials over the computational domain (p–refinement), which in combination with h–refinement leads to so–called hp–adaptivity.

Mesh adaptation in finite element discretizations should be based on rigorous *a posteriori* error estimates; for hyperbolic/nearly–hyperbolic equations such estimates should reflect the inherent mechanisms of error propagation (see [70, 76]). These considerations are particularly important when local quantities such as point values, local averages or flux integrals of the analytical solution are to be computed with high accuracy. In the context of aerodynamic flow simulations, it is of vital importance that certain force coefficients, such as the drag, lift and moment on a body immersed within a compressible fluid, are reliably and efficiently computed. Selective error estimates of this kind can be obtained by the optimal control technique proposed in [36] and [23] which is based on duality arguments analogous to those from the *a priori* error analysis of finite element methods. In the resulting *a posteriori* error estimates the element-residuals of the computed solution are multiplied by local weights involving the adjoint solution. These weights represent the sensitivity of the relevant error quantity with respect to variations of the local mesh size. Since the adjoint solution is usually unknown analytically, it has to be approximated numerically. On the basis of the resulting *a posteriori* error estimate the current mesh is locally adapted and then new approximations to the primal and adjoint solution are computed. This feed-back process is repeated, for instance, until the required error tolerance is reached. In this way, optimal meshes, or in the *hp*–setting, optimal finite element spaces can be obtained for various kinds of error measures, where *optimal* can mean *most economical for achieving a prescribed accuracy* TOL or *most accurate for a given maximum number* N_{max} *of degrees of freedom.* This approach is quite universal as it can, in principle, be applied to almost any problem, linear or nonlinear, as long as it is posed in a variational setting.

This lecture course covers the theory of so-called duality-based *a posteriori* error estimation of DG finite element methods. In particular, we formulate consistent and adjoint consistent DG methods for the numerical approximation of both the compressible Euler and Navier–Stokes equations; in the latter case, the viscous terms are discretized based on employing an interior penalty method. By exploiting a duality argument, adjoint–based *a posteriori* error indicators will be established. Moreover, application of these computable bounds within automatic adaptive finite element algorithms will be developed. Here, a variety of isotropic and anisotropic adaptive strategies, as well as *hp*–mesh refinement will be investigated.

The outline of these notes is as follows. In Section 2 we give an introduction to the adjoint–based *a posteriori* error estimation and mesh refinement for linear problems, and their subsequent exploitation within an automatic adaptive finite element algorithms. Then, in Section 3 we introduce both the compressible Euler and Navier–Stokes equations and formulate DG numerical methods for their discretization. In particular, here we will be concerned with the derivation of so-called adjoint consistent methods, which ensure the optimal approximation of target functionals of the underlying solution. Section 4 is devoted to the derivation of adjoint–based *a posteriori* error bounds for the computed error in a given target functional of interest. Moreover, extensions to the case when there are multiple quantities of interest will be considered. The practical performance of these *a posteriori* error estimates within adaptive finite element algorithms will be studied through a series of numerical experiments. In Section 5 we consider the generalization of the above ideas to the case when anisotropic mesh refinement is permitted. In this setting, we derive both *a priori* and *a posteriori* error bounds for the DG approximation of linear functionals of the underlying analytical solution. The *a priori* analysis is fully explicit in terms of the anisotropy of the underlying computational mesh. Further, we introduce an anisotropic refinement algorithm, based on

choosing the most competitive subdivision of a given element from a series of trial (Cartesian) refinements. The extension of these ideas to general anisotropic *hp*–version DG finite element methods is undertaken in Section 6. Finally, Section 7 is devoted to the application of goal-oriented adaptive finite element algorithms to complex aerodynamic flows, including three dimensional laminar flows as well as two and three dimensional turbulent flows.

Before we embark on Section 2, we first take a brief excursion into the theory of function spaces to introduce the notational conventions used throughout these lecture notes.

1.1 Elements of function space theory

The aim of this section is to provide a brief overview of some elementary results from the theory of function spaces and to introduce the notation which will be used throughout. For proofs and further technical details on classical function spaces the reader is referred to the monograph of Adams [1], for example.

1.1.1 Spaces of continuous functions

Let $\mathbb{N}$ denote the set of all nonnegative integers. An n–tuple $\alpha = (\alpha_1, \ldots, \alpha_n)$ in $\mathbb{N}^n$ will be referred to as a *multi-index*; the nonnegative integer $|\alpha| = |\alpha_1| + \cdots + |\alpha_n|$ is called the length of the multi-index α. We define $\partial^\alpha = \partial_1^{\alpha_1} \ldots \partial_n^{\alpha_n}$, where $\partial_j = \partial/\partial x_j$ for $j = 1, \ldots, n$.

Suppose that ω is an open set in $\mathbb{R}^n$. For $k \in \mathbb{N}$, we denote by $C^k(\omega)$ the set of all continuous real-valued functions u defined on ω such that $\partial^\alpha u$ is continuous on ω for every multi-index α, $|\alpha| \leq k$. When $k = 0$, we shall write $C(\omega)$ in lieu of $C^0(\omega)$. For $k = \infty$, $C^\infty(\omega)$ will denote the intersection $\bigcap_{k \geq 0} C^k(\omega)$.

We shall also require spaces of functions defined over the closure $\bar{\omega}$ of an open set $\omega \subset \mathbb{R}^d$. For $k \in \mathbb{N}$, $C^k(\bar{\omega})$ will signify the set of all $u \in C^k(\omega)$ such that $\partial^\alpha u$ can be continuously extended from ω onto $\bar{\omega}$ for every multi-index α, $|\alpha| \leq k$. Further, we define $C^\infty(\bar{\omega})$ as the intersection $\bigcap_{k \geq 0} C^k(\bar{\omega})$. The notation $C^0(\bar{\omega})$ is abbreviated to $C(\bar{\omega})$.

Assuming that ω is a *bounded* open set in $\mathbb{R}^n$ and $k \in \mathbb{N}$, the linear space $C^k(\bar{\omega})$ is a Banach space equipped with the norm

$$\|u\|_{C^k(\bar{\omega})} = \max_{|\alpha| \leq k} \sup_{\mathbf{x} \in \omega} |\partial^\alpha u(\mathbf{x})| \,.$$

For $k \in \mathbb{N}$ we denote by $C^{k,1}(\bar{\omega})$ the set of all $u \in C^k(\bar{\omega})$ such that the quantity

$$|u|_{C^{k,1}(\bar{\omega})} = \max_{|\alpha| = k} \sup_{\mathbf{x} \neq \mathbf{y},\ \mathbf{x}, \mathbf{y} \in \omega} \frac{|\partial^\alpha u(\mathbf{x}) - \partial^\alpha u(\mathbf{y})|}{|\mathbf{x} - \mathbf{y}|}$$

is finite. $C^{k,1}(\bar{\omega})$ is a Banach space with the norm

$$\|u\|_{C^{k,1}(\bar{\omega})} = \|u\|_{C^k(\bar{\omega})} + |u|_{C^{k,1}(\bar{\omega})} \,.$$

Clearly, $C^{k+1}(\bar{\omega}) \subset C^{k,1}(\bar{\omega})$. When u belongs to $C^{0,1}(\bar{\omega})$, it is said to be *Lipschitz continuous* on $\bar{\omega}$.

The *support*, $\operatorname{supp} u$, of a continuous function u defined on an open set ω is the closure in ω of the set $\{\mathbf{x} \in \omega \ : \ u(\mathbf{x}) \neq 0\}$; in other words, $\operatorname{supp} u$ is the smallest closed subset of ω such that $u = 0$ on $\omega \setminus \operatorname{supp} u$. For $k = 0, 1, \ldots, \infty$, $C_0^k(\omega)$ denotes the set of all $u \in C^k(\omega)$ whose support is a bounded (and, by definition, closed) subset of ω.

1.1.2 Spaces of integrable functions

For $p \geq 1$ and an open set $\omega \subset \mathbb{R}^n$, $L_p(\omega)$ will denote the set of all real-valued Lebesgue measurable functions u defined on ω such that $|u|^p$ is integrable on ω with respect to the Lebesgue measure $\mathrm{d}\mathbf{x} = \mathrm{d}x_1 \ldots \mathrm{d}x_n$; it is implicitly assumed that any two functions which are equal almost everywhere (i.e., equal, except maybe on a set of zero Lebesgue measure) are identified. $L_p(\omega)$ is a Banach space equipped with the norm

$$\|u\|_{L_p(\omega)} = \left(\int_\omega |u(\mathbf{x})|^p \, \mathrm{d}\mathbf{x} \right)^{1/p} .$$

When $p = 2$, $L_2(\omega)$ is a Hilbert space with the inner product

$$(u, v)_\omega = \int_\omega u(\mathbf{x})\, v(\mathbf{x}) \, \mathrm{d}\mathbf{x} .$$

In the case when $\omega \equiv \Omega$, we write $(\cdot, \cdot)$ in lieu of $(\cdot, \cdot)_\Omega$.

$L_\infty(\omega)$ denotes the set of all real-valued Lebesgue measurable functions u defined on ω such that $|u|$ has finite essential supremum; the essential supremum of $|u|$ is defined as the infimum of the set of all positive real numbers M such that $|u| \leq M$ almost everywhere on ω. Again, any two functions that are equal almost everywhere on ω are identified. $L_\infty(\omega)$ is a Banach space with norm

$$\|u\|_{L_\infty(\omega)} = \mathrm{ess.sup}_{\mathbf{x}\in\omega} |u(\mathbf{x})| .$$

Hölder's Inequality. Let $u \in L_p(\omega)$ and $v \in L_q(\omega)$, where $1/p + 1/q = 1$, $1 \leq p, q \leq \infty$. Then $uv \in L_1(\omega)$ and

$$\left| \int_\omega u(\mathbf{x})\, v(\mathbf{x}) \, \mathrm{d}\mathbf{x} \right| \leq \|u\|_{L_p(\omega)} \|v\|_{L_q(\omega)} .$$

In the special case when $p = q = 2$, this inequality is referred to as the *Cauchy-Schwarz Inequality.*

1.1.3 Sobolev spaces

Given that ω is an open set in $\mathbb{R}^n$, k a non-negative integer and $1 \leq p \leq \infty$, we define the *Sobolev space*

$$W_p^k(\omega) = \{ u \in L_p(\omega) \, : \, \partial^\alpha u \in L_p(\omega) , \quad |\alpha| \leq k \} ,$$

and equip it with the *Sobolev norm* defined by

$$\|u\|_{W_p^k(\omega)} = \begin{cases} \left(\sum_{|\alpha|\leq k} \|\partial^\alpha u\|_{L_p(\omega)}^p \right)^{1/p} , & \text{if } 1 \leq p < \infty , \\ \\ \|u\|_{W_\infty^k(\omega)} = \max_{|\alpha|\leq k} \|\partial^\alpha u\|_{L_\infty(\omega)} , & \text{if } p = \infty . \end{cases}$$

The associated *Sobolev seminorm* is defined by

$$|u|_{W_p^k(\omega)} = \begin{cases} \left(\sum_{|\alpha|=k} \|\partial^\alpha u\|_{L_p(\omega)}^p \right)^{1/p} , & \text{if } 1 \leq p < \infty , \\ \\ \max_{|\alpha|=k} \|\partial^\alpha u\|_{L_\infty(\omega)} , & \text{if } p = \infty . \end{cases}$$

In these definitions the derivatives are to be understood in the sense of distributions. The Sobolev space $W_p^k(\omega)$ is a Banach space with the norm $\|\cdot\|_{W_p^k(\omega)}$, $1 \le p \le \infty$, $k \ge 0$. Specifically, for $p = 2$, the normed linear space $W_2^k(\omega)$ is a Hilbert space with the inner product

$$(u, v)_{W_2^k(\omega)} = \sum_{|\alpha| \le k} (\partial^\alpha u, \partial^\alpha v)_\omega \,,$$

where $(\cdot, \cdot)_\omega$ denotes the inner product in $L_2(\omega)$.

Finer smoothness properties of integrable functions can be detected by considering fractional-order Sobolev spaces. Given that s is a positive real number, $s \notin \mathbb{N}$, let us write $s = m + \sigma$, where $0 < \sigma < 1$ and $m = [s]$ is the integer part of s. The fractional-order Sobolev space $W_p^s(\omega)$, $1 \le p < \infty$, is the set of all $u \in W_p^m(\omega)$ such that

$$|u|_{W_p^s(\omega)} = \left\{ \sum_{|\alpha|=m} \int_\omega \int_\omega \frac{|D^\alpha u(\mathbf{x}) - D^\alpha u(\mathbf{y})|^p}{|\mathbf{x} - \mathbf{y}|^{n+\sigma p}} \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{y} \right\}^{1/p} < \infty \,,$$

with the usual modification when $p = \infty$. The fractional-order Sobolev norm of index s is defined by

$$\|u\|_{W_p^s(\omega)} = \begin{cases} \left\{ \|u\|^p_{W_p^m(\omega)} + |u|^p_{W_p^s(\omega)} \right\}^{1/p}, & \text{if } 1 \le p < \infty \,, \\ \\ \|u\|_{W_\infty^m(\omega)} + |u|_{W_\infty^s(\omega)} \,, & \text{if } p = \infty \,. \end{cases}$$

The fractional-order Sobolev space $W_p^s(\omega)$ is a Banach space with this norm.

When $p = 2$ we shall write H^s in place of W_2^s to signify the fact that we are dealing with a Hilbert space. We denote by $H_0^s(\omega)$ the closure of $C_0^\infty(\omega)$ in the norm of $H^s(\omega)$; when ω is a Lipschitz domain and $1/2 < s < 3/2$, this space coincides with the set of all those functions in $H^s(\omega)$ whose trace on $\partial\omega$ is equal to zero.

2 Motivation: Linear problems and adjoint equations

In this section we present an overview of the general theoretical framework of adjoint–based *a posteriori* error estimation developed by C. Johnson and R. Rannacher and their collaborators. For a detailed discussion, we refer to the series of articles [23, 36, 71, 79], and the references cited therein. To this end, we introduce the *a posteriori* error estimation for linear problems in Section 2.1. Then in Section 2.2 we give a framework for deriving adjoint problem for linear primal problems. This is then applied to the linear advection equation in Section 2.3. A numerical example in Section 2.4 highlights the practical importance of adjoint–based refinement.

2.1 Error estimation for linear problems

We begin by considering a linear problem

$$Lu = f \quad \text{in } \Omega, \qquad\qquad Bu = g \quad \text{on } \Gamma, \tag{1}$$

where $f \in L_2(\Omega)$, $g \in L_2(\Gamma)$, L denotes a linear differential operators on Ω, and B denotes a linear boundary operator on Γ.

Let the linear problem (1) be discretized as follows: Find $u_h \in V_{h,p}$ such that

$$\mathcal{B}(u_h, v_h) = \ell(v_h) \qquad \forall v_h \in V_{h,p}, \tag{2}$$

where $V_{h,p}$ is a finite element space consisting of piecewise polynomial functions of degree p on a partition $\mathcal{T}_h$ of the domain Ω in elements $\kappa \in \mathcal{T}_h$ of size h. Furthermore, $\mathcal{B}(\cdot,\cdot) : V \times V \to \mathbb{R}$ is a bilinear form and $l(\cdot) : V \to \mathbb{R}$ is a linear form including the forcing function f and the boundary value function g. Here, V is some suitably chosen function space including the analytical solution $u \in V$ to the primal problem and satisfying $V_{h,p} \subset V$.

Furthermore, let us assume that the discretization (2) is *consistent*, i.e., the analytical solution $u \in V$ satisfies the following equation:

$$\mathcal{B}(u, v) = \ell(v) \qquad \forall v \in V. \tag{3}$$

In many problems of physical importance the quantities of interest may be a series of target or error functionals $J_i(\cdot)$, $i = 1, \ldots, N$, $N \geq 1$, of the solution. Relevant examples include the mean value of the solution, the mean flow across a line, the point value of the solution or different scalar quantities which can be computed from the solution u. For compressible flows, which are not covered by the theory in this introductory section, such a quantity $J(u)$ could represent an aerodynamic force coefficient, like the drag, lift or moment coefficient. For simplicity, we restrict ourselves to the case of a single *linear* target functional, i.e., $N = 1$, and write $J(\cdot) \equiv J_1(\cdot)$; for the extension of the proceeding theory to multiple target functionals, see Section 4.3; cf., also, [60]. In order to obtain a computable *a posteriori* bound on the error between the true value of the functional $J(u)$ and the computed value $J(u_h)$, we begin by noting the Galerkin orthogonality of the discretization (2):

$$\mathcal{B}(u, v_h) - \mathcal{B}(u_h, v_h) = \mathcal{B}(u - u_h, v_h) = 0 \quad \forall v_h \in V_{h,p}. \tag{4}$$

This will be a key ingredient in the following *a posteriori* error analysis.

We now introduce the following *adjoint* problem: find $z \in V$ such that

$$\mathcal{B}(w, z) = J(w) \quad \forall w \in V; \tag{5}$$

We assume that (5) possesses a unique solution; clearly, the validity of this assumption depends on both the definition of $\mathcal{B}(\cdot, \cdot)$ and the choice of the functional under consideration. Important examples which are covered by our hypothesis are discussed below, cf. [77].

For the proceeding error analysis, we must therefore *assume* that the adjoint problem (5) is well–posed. Under this assumption, employing the Galerkin orthogonality property (4) we deduce the following error representation formula:

$$\begin{aligned} J(u) - J(u_h) &= J(u - u_h) = \mathcal{B}(u - u_h, z) \\ &= \mathcal{B}(u - u_h, z - z_h) \\ &= \ell(z - z_h) - \mathcal{B}(u_h, z - z_h) \end{aligned} \tag{6}$$

for all z_h in the finite element space $V_{h,p}$. On the basis of the general error representation formula (6), *a posteriori* estimates which provide upper bounds on the true error in the computed target functional $J(\cdot)$ may be deduced. The simplest approach is to first decompose the right–hand side of (6) as a summation of local error indicators η_κ over the elements κ in the computational mesh $\mathcal{T}_h$, i.e., we write

$$J(u) - J(u_h) = \ell(z - z_h) - \mathcal{B}(u_h, z - z_h) \equiv \mathcal{R}(u_h, z - z_h) = \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa; \tag{7}$$

then, upon application of the triangle inequality, we deduce the following weighted *a posteriori* error bound.

Theorem 2.1 *Let u and u_h denote the solutions of* (1) *and* (2)*, respectively, and suppose that the adjoint problem (5) is well–posed. Then, the following* a posteriori *error bound holds:*

$$|J(u) - J(u_h)| \leq \mathcal{R}_{|\Omega|}(u_h, z - z_h) \equiv \sum_{\kappa \in \mathcal{T}_h} |\eta_\kappa| \tag{8}$$

for all z_h in $V_{h,p}$.

We remark that the local error indicators η_κ appearing on the right–hand side of (9) involve the multiplication of finite element *residuals* depending only on u_h with local weighting terms involving the difference between the adjoint solution z satisfying (5) and its projection/interpolant z_h onto the finite element space $V_{h,p}$. These weights represent the sensitivity of the error in the target functional $J(\cdot)$ with respect to variations of the local element residuals; indeed, they provide invaluable information concerning the global transport of the error, which is essential for efficient error control.

Since the solution to the adjoint problem is usually unknown analytically it must be numerically approximated, cf. [23, 43, 58]. Replacing the unknown exact adjoint solution z in (7) by a numerical approximation $\bar{z}_h \notin V_{h,p}$, we obtain following approximate error representation

$$J(u) - J(u_h) = \mathcal{R}(u_h, z - z_h) \approx \mathcal{R}(u_h, \bar{z}_h - z_h) = \sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa. \tag{9}$$

Note that the so-called *adjoint-based indicators* η_κ in (9) can be used to drive an adaptive algorithm targeted at the accurate and efficient approximation of the target quantity $J(u)$.

In the following sections we give some examples of adjoint problems.

2.2 Derivation of adjoint problems for linear primal problems

Let us again consider the linear PDE problem (1). Furthermore, let $J(\cdot)$ be a linear target functional given by

$$J(u) = (j_\Omega, u) + (j_\Gamma, Cu)_\Gamma \equiv \int_\Omega j_\Omega \, u \, \mathrm{d}\mathbf{x} + \int_\Gamma j_\Gamma \, Cu \, \mathrm{d}s, \tag{10}$$

where $j_\Omega \in L_2(\Omega)$, $j_\Gamma \in L_2(\Gamma)$, C is an operator on Γ which may be differential, and $(\cdot,\cdot)$ and $(\cdot,\cdot)_\Gamma$ denote the $L_2(\Omega)$ and $L_2(\Gamma)$ inner products, respectively. We assume that the target functional (10) is *compatible* with the primal problem (1), i.e., we assume that there are linear operators L^*, B^* and C^* such that following *compatibility condition* holds:

$$(Lu, z) + (Bu, C^*z)_\Gamma = (u, L^*z) + (Cu, B^*z)_\Gamma. \tag{11}$$

Then, L^*, B^* and C^* are the so-called *adjoint operators* of L, B and C, respectively. We note that for given operators L and B associated with the primal problem (1) only a subset of possible target functionals (10) with operators C are compatible; indeed, many definitions of functionals may fail to satisfy the compatibility condition (11). However, *assuming* that (11) holds, the adjoint problem associated to (1) and (10) is given by

$$L^*z = j_\Omega \quad \text{in } \Omega, \qquad\qquad B^*z = j_\Gamma \quad \text{on } \Gamma. \tag{12}$$

Remark 2.2 *In an adjoint-based optimization framework, see e.g. [47], this ensures that*

$$\begin{aligned} J(u) &= (u, j_\Omega) + (Cu, j_\Gamma)_\Gamma = (u, L^*z) + (Cu, B^*z)_\Gamma \\ &= (Lu, z) + (Bu, C^*z)_\Gamma = (f, z) + (g, C^*z)_\Gamma. \end{aligned} \tag{13}$$

2.3 The linear advection equation and adjoint problems

The first model problem we consider is the linear advection equation: find u such that

$$\nabla \cdot (\mathbf{b}u) + cu = f \quad \text{in } \Omega, \qquad\qquad u = g \quad \text{on } \Gamma_-, \tag{14}$$

where $f \in L_2(\Omega)$, $\mathbf{b} \in [C^1(\Omega)]^d$, $c \in L_\infty(\Omega)$ and $g \in L_2(\Gamma_-)$, where

$$\Gamma_- = \{\mathbf{x} \in \Gamma, \mathbf{b}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0\} \tag{15}$$

denotes the inflow part of the boundary $\Gamma = \partial\Omega$. In the following, by $\Gamma_+ = \Gamma \setminus \Gamma_-$ we denote the outflow boundary. In order to derive the continuous adjoint problem, we multiply the left hand side of (14) by z, integrate over Ω and perform integration by parts. Thereby, we obtain

$$(\nabla \cdot (\mathbf{b}u) + cu, z) + (u, -\mathbf{b} \cdot \mathbf{n}\, z)_{\Gamma_-} = (u, -\mathbf{b} \cdot \nabla z + cz) + (u, \mathbf{b} \cdot \mathbf{n}\, z)_{\Gamma_+}. \tag{16}$$

Comparing with (11), we see that for $Lu \equiv \nabla \cdot (\mathbf{b}u) + cu$ in Ω and

$$\begin{aligned} &Bu = u, && Cu = 0, && \text{on } \Gamma_-, \\ &Bu = 0, && Cu = u, && \text{on } \Gamma_+, \end{aligned}$$

the adjoint operators are given by $L^*z \equiv -\mathbf{b}\cdot\nabla z + cz$ in Ω and

$$\begin{aligned} &B^*z = 0, && C^*z = -\mathbf{b}\cdot\mathbf{n}\, z && \text{on } \Gamma_-,\\ &B^*z = \mathbf{b}\cdot\mathbf{n}\, z, && C^*z = 0 && \text{on } \Gamma_+. \end{aligned}$$

In particular, for

$$J(u) = \int_\Omega j_\Omega\, u\,\mathrm{d}\mathbf{x} + \int_\Gamma j_\Gamma\, Cu\,\mathrm{d}s = \int_\Omega j_\Omega\, u\,\mathrm{d}\mathbf{x} + \int_{\Gamma_+} j_\Gamma\, u\,\mathrm{d}s, \tag{17}$$

the continuous adjoint problem is given by

$$-\mathbf{b}\cdot\nabla z + cz = j_\Omega \quad \text{in } \Omega, \tag{18}$$

subject to the boundary condition

$$\mathbf{b}\cdot\mathbf{n}\, z = j_\Gamma \quad \text{on } \Gamma_+. \tag{19}$$

In the following, we give some examples:

1. *Outflow normal flux:* Given a weight function $\psi \in L_2(\Gamma_+)$ and setting $j_\Omega \equiv 0$ and $j_\Gamma = \mathbf{b}\cdot\mathbf{n}\,\psi$ in (18) and (19) we obtain the weighted normal flux through the outflow boundary Γ_+ defined by
$$J(u) = \int_{\Gamma_+} \mathbf{b}\cdot\mathbf{n}\,\psi\, u\,\mathrm{d}s.$$
Then, z is the unique solution to the following boundary value problem: find z such that
$$\begin{aligned} -\mathbf{b}\cdot\nabla z + cz &= 0 \quad && \text{in } \Omega,\\ z &= \psi \quad && \text{on } \Gamma_+. \end{aligned}$$

2. *Mean value:* Given a weight function $j_\Omega \in L_2(\Omega)$ and setting $j_\Gamma \equiv 0$ in (19) we obtain the weighted mean value given by
$$J(u) = \int_\Omega j_\Omega\, u\,\mathrm{d}\mathbf{x}.$$
In this case, z is the solution to following adjoint problem: find z such that
$$\begin{aligned} -\mathbf{b}\cdot\nabla z + cz &= j_\Omega \quad && \text{in } \Omega,\\ z &= 0 \quad && \text{on } \Gamma_+. \end{aligned}$$

3. *Point value:* Under the assumption that the analytical solution u is a continuous function in the neighbourhood of a given point $\mathbf{x}_0 \in \Omega$ we consider the evaluation of the point value
$$J(u) = u(\mathbf{x}_0).$$
Then, z is the solution to following adjoint problem: find z such that
$$\begin{aligned} -\mathbf{b}\cdot\nabla z + cz &= \delta_{\mathbf{x}_0} \quad && \text{in } \Omega,\\ z &= 0 \quad && \text{on } \Gamma_+, \end{aligned}$$

where $\delta_{\mathbf{x}_0}$ denotes a δ-distribution at the point $\mathbf{x}_0$ with the property

$$\int_\Omega \delta_{\mathbf{x}_0} u \, \mathrm{d}\mathbf{x} = u(\mathbf{x}_0).$$

In this setting the weak solution of the adjoint problem is a measure rather than a regular distribution; in particular, z does not belong to $L_2(\Omega)$. Thus, to avoid technical complications, we may mollify the functional J by considering a nonnegative function φ in $L_{1,\mathrm{loc}}(\mathbb{R}^d)$ whose support is contained in the unit ball $B(0,1)$ centered at $\mathbf{x} = \mathbf{0}$ and such that the integral of φ over $B(0,1)$ is equal to 1. Writing $\psi(\mathbf{x}) = \varphi_\varepsilon(\mathbf{x}) \equiv \varepsilon^{-d}\,\varphi((\mathbf{x}-\mathbf{x}_0)/\varepsilon)$, the mollified functional

$$J_M(u) = \int_\Omega \psi \, u \, \mathrm{d}\mathbf{x}$$

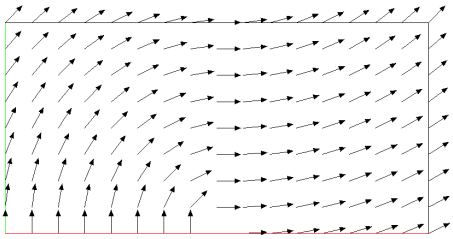
converges to $u(\mathbf{x}_0)$ as $\varepsilon \to 0$. Further, setting $J(w) = J_M(w)$ into (5) as the right-hand side, for $0 \ll \varepsilon < 1$ fixed, now results in a unique solution z.

The adjoint problem transports information along the characteristics of the primal problem but in the opposite direction. The solution to the adjoint problem is related to the domain of influence for the target quantity under consideration, in the sense that the solution at all points within the support of the adjoint solution may affect the value of the target quantity. From (6) we see that the error $J(u) - J(u_h)$ of the discrete solution u_h measured in terms of the target quantity depends on the residuals of the primal solution within this domain while the adjoint solution traces back to the origin of these residuals. In fact, the adjoint solution describes *quantitatively* to what extent the residuals contribute to the error in the target quantity. This information can be used by an adaptive algorithm that equilibrates the adjoint-based indicators η_κ, see (9), by refining and coarsening the mesh. Such an algorithm leads to meshes where the elements are distributed in order to reduce the contributions to the error in the target quantity.

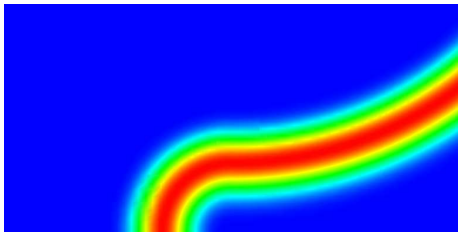
2.4 Numerical example: Linear advection equation

As first numerical example taken from [51, 52] we consider the linear advection equation (14) on $\Omega = [0,2] \times [0,1] \in \mathbb{R}^2$ with a vector field $\mathbf{b}$ as shown in Figure 1(a). For this problem and the prescribed boundary values on the inflow boundary ($u(x,0) = 1$ for $\frac{1}{8} < x < \frac{3}{4}$ and zero boundary values elsewhere) the solution is shown in Figure 1(c). Here, the two jumps of the discontinuous boundary function are transported along the characteristic directions given by the vector field. Assume that we are interested in the values of the solution on the part $\frac{1}{4} < y < 1$ of the right outflow boundary. Let us take, for example, $J(u) = \int_{\Gamma_+} u\psi \, ds$ as target functional, where ψ is chosen to be very smooth, $\psi(2,y) = \exp\left(\left(\frac{3}{8}\right)^{-2} - \left((y-\frac{5}{8})^2 - \frac{3}{8}\right)^{-2}\right)$ for $\frac{1}{4} < y < 1$ and 0 elsewhere, such that also the corresponding adjoint solution is smooth, see Figure 1(b).

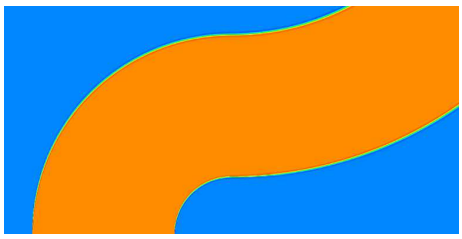
Figure 1(d) shows the numerical solution on the adaptively refined mesh, see Figure 1(e), which has been refined using the adjoint-based indicators. Note that the refinement takes place at the position of only one of the discontinuities present in u. Indeed, the second discontinuity is not resolved at all, as it is outside of the support of the adjoint solution,



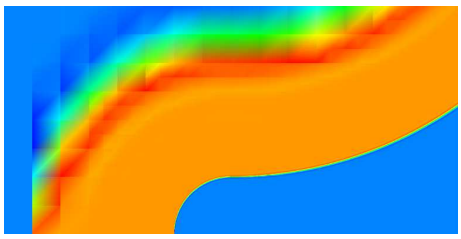
(a) Vector field $\mathbf{b}$



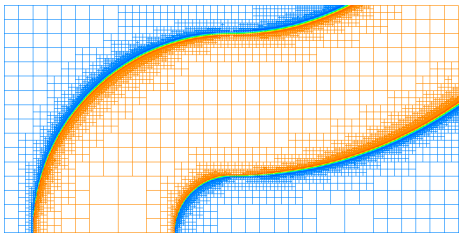
(b) Adjoint solution related to the target functional $\int_{\partial\Omega^+} u\psi \, ds$ with smooth ψ



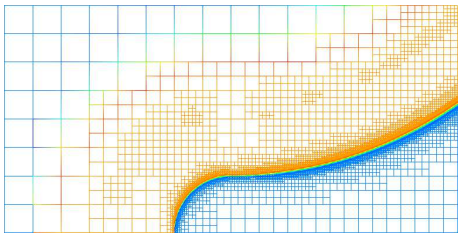
(c) Primal solution on mesh, see Figure 1(e)



(d) Primal solution on mesh, see Figure 1(f)



(e) Traditionally refined mesh



(f) Adjoint-based refined mesh

Figure 1: Linear advection equation: Comparison of traditional and adjoint-based indicators, [51, 52].

and hence does not belong to the domain of influence of the target quantity. Thereby, the residuals in the neighborhood of this discontinuity do *not* contribute to the error in the target quantity. Comparing the two meshes, Figure 1(e) and Figure 1(f), it is obvious that the mesh in Figure 1(f) is more cost-efficient for evaluating the value of the target quantity than the mesh refined with traditional residual–based indicators which do not include the adjoint solution.

3 Discontinuous Galerkin methods for compressible flows and their corresponding adjoint problems

In this section we introduce the compressible Euler and Navier-Stokes equations. We then derive the corresponding adjoint problems connected to specific target quantities. To this end, the derivation of adjoint problems as outlined in Section 2.2 for linear problems will be extended to nonlinear problems. Furthermore, we will introduce the DG discretization of the compressible Euler and Navier-Stokes equations which is both consistent and adjoint consistent.

3.1 The compressible Euler equations

The compressible Euler equations are a nonlinear system of conservation equations (conservation of mass, momentum and energy) describing inviscid compressible flows which are frequently used as a simple model for gas flows. In particular, here we consider the stationary Euler equations

$$\nabla \cdot \mathcal{F}^c(\mathbf{u}) = 0 \quad \text{in } \Omega, \tag{20}$$

where Ω is a bounded open Lipschitz domain in $\mathbb{R}^3$. The vector of conservative variables $\mathbf{u}$ is given by $\mathbf{u} = (\rho, \rho v_1, \rho v_2, \rho v_3, \rho E)^\top$ and the convective flux $\mathcal{F}^c(\mathbf{u}) = (\mathbf{f}_1^c(\mathbf{u}), \mathbf{f}_2^c(\mathbf{u}), \mathbf{f}_3^c(\mathbf{u}))^\top$ in three dimensions is given by

$$\mathbf{f}_1^c(\mathbf{u}) = \begin{bmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ \rho v_1 v_3 \\ \rho H v_1 \end{bmatrix}, \quad \mathbf{f}_2^c(\mathbf{u}) = \begin{bmatrix} \rho v_2 \\ \rho v_2 v_1 \\ \rho v_2^2 + p \\ \rho v_2 v_3 \\ \rho H v_2 \end{bmatrix}, \text{ and } \mathbf{f}_3^c(\mathbf{u}) = \begin{bmatrix} \rho v_3 \\ \rho v_3 v_1 \\ \rho v_3 v_2 \\ \rho v_3^2 + p \\ \rho H v_3 \end{bmatrix}, \tag{21}$$

where ρ, $\mathbf{v} = (v_1, v_2, v_3)^\top$, p and E denote the density, velocity vector, pressure and specific total energy, respectively. Additionally, H is the total enthalpy given by

$$H = E + \frac{p}{\rho} = e + \tfrac{1}{2}\mathbf{v}^2 + \frac{p}{\rho}, \tag{22}$$

where e is the specific static internal energy, and the pressure is determined by the equation of state of an ideal gas

$$p = (\gamma - 1)\rho e, \tag{23}$$

where $\gamma = c_p/c_v$ is the ratio of specific heat capacities at constant pressure, c_p, and constant volume, c_v; for dry air, $\gamma = 1.4$. The flux Jacobians $A_i(\mathbf{u}) := \partial_{\mathbf{u}} \mathbf{f}_i^c(\mathbf{u})$, $i = 1, 2, 3$, are given by

$$A_1(\mathbf{u}) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -v_1^2 + \frac{1}{2}(\gamma-1)\mathbf{v}^2 & (3-\gamma)v_1 & -(\gamma-1)v_2 & -(\gamma-1)v_3 & \gamma - 1 \\ -v_1 v_2 & v_2 & v_1 & 0 & 0 \\ -v_1 v_3 & v_3 & 0 & v_1 & 0 \\ v_1\left(\frac{1}{2}(\gamma-1)\mathbf{v}^2 - H\right) & H - (\gamma-1)v_1^2 & -(\gamma-1)v_1 v_2 & -(\gamma-1)v_1 v_3 & \gamma v_1 \end{pmatrix},$$

$$A_2(\mathbf{u}) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ -v_1v_2 & v_2 & v_1 & 0 & 0 \\ -v_2^2 + \frac{1}{2}(\gamma-1)\mathbf{v}^2 & -(\gamma-1)v_1 & (3-\gamma)v_2 & -(\gamma-1)v_3 & \gamma-1 \\ -v_2v_3 & 0 & v_3 & v_2 & 0 \\ v_2\left(\frac{1}{2}(\gamma-1)\mathbf{v}^2 - H\right) & -(\gamma-1)v_1v_2 & H-(\gamma-1)v_2^2 & -(\gamma-1)v_2v_3 & \gamma v_2 \end{pmatrix}.$$

$$A_3(\mathbf{u}) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ -v_1v_3 & v_3 & 0 & v_1 & 0 \\ -v_2v_3 & 0 & v_3 & v_2 & 0 \\ -v_3^2 + \frac{1}{2}(\gamma-1)\mathbf{v}^2 & -(\gamma-1)v_1 & -(\gamma-1)v_2 & (3-\gamma)v_3 & \gamma-1 \\ v_3\left(\frac{1}{2}(\gamma-1)\mathbf{v}^2 - H\right) & -(\gamma-1)v_1v_3 & -(\gamma-1)v_2v_3 & H-(\gamma-1)v_3^2 & \gamma v_3 \end{pmatrix}.$$

Writing $\mathbf{n} \in \mathbb{R}^3$ to denote the unit outward normal vector to the boundary $\Gamma = \partial\Omega$, the normal flux Jacobian $A_n(\mathbf{u}, \mathbf{n})$ is given by

$$A_n(\mathbf{u}, \mathbf{n}) = \sum_{i=1}^{3} n_i A_i(\mathbf{u}). \tag{24}$$

Moreover, the eigenvalues of the matrix $A_n(\mathbf{u}, \mathbf{n})$ are

$$\lambda_1 = \mathbf{v}\cdot\mathbf{n} - c, \quad \lambda_2 = \lambda_3 = \lambda_4 = \mathbf{v}\cdot\mathbf{n}, \quad \lambda_5 = \mathbf{v}\cdot\mathbf{n} + c, \tag{25}$$

where $c = \sqrt{\gamma p/\rho}$ denotes the speed of sound.

The system of conservation equations (20) must be supplemented by appropriate boundary conditions; for example at inflow/outflow boundaries, we require that

$$A_n^-(\mathbf{u}, \mathbf{n})\,(\mathbf{u} - \mathbf{g}) = 0 \quad \text{on } \Gamma \tag{26}$$

where $\mathbf{g}$ is a (given) vector function. Here, $A_n^{\pm}(\mathbf{u}, \mathbf{n})$ denotes the positive/negative part of $A_n(\mathbf{u}, \mathbf{n})$ defined by

$$A_n^{\pm}(\mathbf{u}, \mathbf{n}) = P\Lambda^{\pm}P^{-1}, \tag{27}$$

where $P = [\mathbf{r}_1, \ldots, \mathbf{r}_5]$ denotes the 5×5 matrix of eigenvectors of $A_n(\mathbf{u}, \mathbf{n})$ and $\Lambda^+ = \mathrm{diag}(\max(\lambda_i, 0))$ and $\Lambda^- = \mathrm{diag}(\min(\lambda_i, 0))$ the 5×5 diagonal matrix of the positive/negative eigenvalues of $A_n(\mathbf{u}, \mathbf{n})$, respectively, with $A_n\mathbf{r}_i = \lambda_i\mathbf{r}_i$, $i = 1, \ldots, 5$.

Considering the signs of λ_i, $i = 1, \ldots, 5$, we distinguish four cases of farfield boundary conditions:

- supersonic inflow: $\lambda_i < 0$, $i = 1, \ldots, 5$,
- subsonic inflow: $\lambda_i < 0$, $i = 1, \ldots, 4$, $\lambda_5 > 0$,
- subsonic outflow: $\lambda_1 < 0$, $\lambda_i > 0$, $i = 2, \ldots, 5$, and
- supersonic outflow: $\lambda_i > 0$, $i = 1, \ldots, 5$.

Each eigenvalue smaller than zero corresponds to an inflow characteristic. The number of variables to be prescribed on the boundary depend on the number of inflow characteristics.

Finally, at wall boundaries we require that the normal velocity vanishes, i.e., $\mathbf{v}\cdot\mathbf{n} = 0$.

3.2 Derivation of adjoint problems for nonlinear primal problems

In this section the derivation of adjoint problems as outlined in Section 2.2 for linear problems will be extended to nonlinear problems. Let us consider following nonlinear problem,

$$Nu = 0 \quad \text{in } \Omega, \qquad\qquad Bu = 0 \quad \text{on } \Gamma, \tag{28}$$

where N is a nonlinear differential (and Fréchet-differentiable) operator and B is a (possibly nonlinear) boundary operator. Let $J(\cdot)$ be a nonlinear target functional of the form

$$J(u) = \int_\Omega j_\Omega(u)\,\mathrm{d}\mathbf{x} + \int_\Gamma j_\Gamma(Cu)\,\mathrm{d}s, \tag{29}$$

with Fréchet derivative

$$J'[u](w) = \int_\Omega j'_\Omega[u]\,w\,\mathrm{d}\mathbf{x} + \int_\Gamma j'_\Gamma[Cu]\,C'[u]w\,\mathrm{d}s, \tag{30}$$

where $j_\Omega(\cdot)$ and $j_\Gamma(\cdot)$ may be nonlinear with derivatives j'_Ω and j'_Γ, respectively, and C is a differential boundary operator on Γ (which may be nonlinear) with derivative C'. Here, $'$ denotes the (total) Fréchet derivative and the square bracket $[\cdot]$ denotes the state about which linearization is performed. Again, we say that the target functional (29) is *compatible* with (28) provided the following compatibility condition holds

$$(N'[u]w, z) + (B'[u]w, (C'[u])^* z)_\Gamma = (w, (N'[u])^* z) + (C'[u]w, (B'[u])^* z)_\Gamma, \tag{31}$$

where $(N'[u])^*$, $(B'[u])^*$ and $(C'[u])^*$ denote the adjoint operators to $N'[u]$, $B'[u]$ and $C'[u]$, respectively. This condition is analogous to (11), with L, B and C replaced by $N'[u]$, $B'[u]$ and $C'[u]$, respectively. *Assuming* that (31) holds the continuous adjoint problem associated to (28) and (30) is:

$$(N'[u])^* z = j'_\Omega[u] \quad \text{in } \Omega, \qquad\qquad (B'[u])^* z = j'_\Gamma[Cu] \quad \text{on } \Gamma. \tag{32}$$

3.3 The adjoint equations to the compressible Euler equations

The most important target quantities in inviscid compressible flows are the pressure induced drag and lift coefficients C_{dp} and C_{lp}, respectively, on a given solid wall boundary $\Gamma_W \subset \Gamma$. These quantities are defined by

$$J(\mathbf{u}) = \int_\Gamma j(\mathbf{u})\,\mathrm{d}s = \frac{1}{C_\infty}\int_{\Gamma_W} p\,\mathbf{n}\cdot\psi\,\mathrm{d}s, \tag{33}$$

where $j(\mathbf{u}) = \frac{1}{C_\infty} p\,\mathbf{n}\cdot\psi$ on Γ_W and $j(\mathbf{u}) \equiv 0$ elsewhere. Here, $C_\infty = \frac{1}{2}\gamma p_\infty M_\infty^2 \bar{A} = \frac{1}{2}\gamma\frac{|\mathbf{v}_\infty|^2}{c_\infty^2} p_\infty \bar{A} = \frac{1}{2}\rho_\infty |\mathbf{v}_\infty|^2 \bar{A}$, where M denotes the Mach number, c the sound speed defined by $c^2 = \gamma p/\rho$, $\bar{A}$ denotes a reference area, and ψ is given by $\psi_d = (\cos(\alpha), 0, \sin(\alpha))^\top$ or $\psi_l = (-\sin(\alpha), 0, \cos(\alpha))^\top$ for the drag and lift coefficient, respectively. Subscripts $_\infty$ indicate free-stream quantities.

In order to derive the continuous adjoint problem, we multiply the left hand side of (20) by $\mathbf{z}$, integrate by parts and linearize about $\mathbf{u}$ to obtain

$$\left(\nabla\cdot\left(\mathcal{F}^c_{\mathbf{u}}[\mathbf{u}](\mathbf{w})\right), \mathbf{z}\right) = -\left(\mathcal{F}^c_{\mathbf{u}}[\mathbf{u}](\mathbf{w}), \nabla\mathbf{z}\right) + \left(\mathbf{n}\cdot\mathcal{F}^c_{\mathbf{u}}[\mathbf{u}](\mathbf{w}), \mathbf{z}\right)_\Gamma, \tag{34}$$

where $\mathcal{F}^c_{\mathbf{u}}[\mathbf{u}] := (\mathcal{F}^c)'[\mathbf{u}]$ denotes the Fréchet derivative of $\mathcal{F}^c$ with respect to $\mathbf{u}$. Here, we already use the subscript $_{\mathbf{u}}$ notation, which we require in Section 3.7 to distinguish from subscript $_{\nabla\mathbf{u}}$ denoting the derivative with respect to $\nabla\mathbf{u}$. Thereby, the variational formulation of the continuous adjoint problem is given by: find $\mathbf{z}$ such that

$$-\left(\mathbf{w}, (\mathcal{F}^c_{\mathbf{u}}[\mathbf{u}])^\top \nabla\mathbf{z}\right) + \left(\mathbf{w}, (\mathbf{n}\cdot\mathcal{F}^c_{\mathbf{u}}[\mathbf{u}])^\top \mathbf{z}\right)_\Gamma = J'[\mathbf{u}](\mathbf{w}) \quad \forall \mathbf{w}\in V. \tag{35}$$

The continuous adjoint problem is then given by

$$-(\mathcal{F}^c_{\mathbf{u}}[\mathbf{u}])^\top \nabla\mathbf{z} = 0 \quad \text{in } \Omega, \qquad\qquad (\mathbf{n}\cdot\mathcal{F}^c_{\mathbf{u}}[\mathbf{u}])^\top \mathbf{z} = j'[\mathbf{u}] \quad \text{on } \Gamma. \tag{36}$$

Using $\mathcal{F}^c(\mathbf{u})\cdot\mathbf{n} = p(0, n_1, n_2, n_3, 0)^\top$ on Γ_W, and the definition of j in (33) we obtain

$$p'[\mathbf{u}](0, n_1, n_2, n_3, 0)\cdot\mathbf{z} = \frac{1}{C_\infty} p'[\mathbf{u}]\mathbf{n}\cdot\boldsymbol{\psi} \quad \text{on } \Gamma_W,$$

which reduces to the boundary condition of the adjoint compressible Euler equations,

$$(B'[\mathbf{u}])^*\mathbf{z} = n_1 z_2 + n_2 z_3 + n_3 z_4 = \frac{1}{C_\infty}\mathbf{n}\cdot\boldsymbol{\psi} \quad \text{on } \Gamma_W. \tag{37}$$

This, in fact, is the adjoint operator of

$$B\mathbf{u} = n_1 u_2 + n_2 u_3 + n_3 u_4 = 0 \quad \text{on } \Gamma_W,$$

which via $n_1u_2 + n_2u_3 + n_3u_4 = \rho(n_1v_1 + n_2v_2 + n_3v_3) = \rho\mathbf{v}\cdot\mathbf{n} = 0$ represents a vanishing normal velocity, $\mathbf{v}\cdot\mathbf{n} = 0$, at wall boundaries.

3.4 DG discretization of the compressible Euler equations

We begin by introducing the necessary notation. Suppose that $\mathcal{T}_h$ is a subdivision of Ω into open element domains κ such that $\bar\Omega = \cup_{\kappa\in\mathcal{T}_h}\bar\kappa$. Let us assume that each $\kappa\in\mathcal{T}_h$ is a smooth bijective image of a fixed reference element $\hat\kappa$, that is, $\kappa = F_\kappa(\hat\kappa)$ for all $\kappa\in\mathcal{T}_h$. On the reference element $\hat\kappa$ we define spaces of polynomials of degree $p\geq 0$ as follows:

$$\mathcal{Q}_p = \operatorname{span}\{\hat{\mathbf{x}}^\alpha : 0\leq\alpha_i\leq p,\ 0\leq i\leq 3\}, \qquad \mathcal{P}_p = \operatorname{span}\{\hat{\mathbf{x}}^\alpha : 0\leq|\alpha|\leq p\}.$$

We now introduce the finite element function space $\mathbf{V}_{h,p}$ consisting of discontinuous vector–valued polynomial functions of degree $p\geq 0$, defined by

$$\begin{aligned}\mathbf{V}_{h,p} = \{\mathbf{v}_h\in[L_2(\Omega)]^5 : \mathbf{v}_h|_\kappa\circ F_\kappa\in[Q_p(\hat\kappa)]^5 \text{ if } \hat\kappa \text{ is the unit hypercube, and}\\ \mathbf{v}_h|_\kappa\circ F_\kappa\in[P_p(\hat\kappa)]^5 \text{ if } \hat\kappa \text{ is the unit simplex}, \kappa\in\mathcal{T}_h\}.\end{aligned} \tag{38}$$

Let κ^+ and κ^- be two adjacent elements of $\mathcal{T}_h$ and $\mathbf{x}$ be an arbitrary point on the interior face $f = \partial\kappa^+\cap\partial\kappa^-$. Given a function vector–valued function $\mathbf{v}$ which is assumed to be smooth inside each element $\kappa^\pm$, by $\mathbf{v}^\pm := \mathbf{v}|_{\partial\kappa^\pm}$ we denote the traces of $\mathbf{v}$ on f taken from within the interior of $\kappa^\pm$, respectively. Indeed, given that κ^+ and κ^- are two adjacent elements of $\mathcal{T}_h$, $\mathbf{v}^\mp$ may be viewed as the *exterior/outer* trace of $\mathbf{v}$ on f relative to $\kappa^\pm$, respectively. For notational simplicity, in the sequel we shall neglect the superscript '+' on the elemental

domain $\kappa \in \mathcal{T}_h$ and write $\mathbf{v}^{\pm}$ to denote the traces of $\mathbf{v}$ on $f \subset \partial\kappa \cap \partial\kappa^-$ taken from within the interior of κ and κ^-, respectively, where κ and κ^- are two adjacent elements of $\mathcal{T}_h$.

For deriving the DG discretization we introduce a weak formulation of (20). In particular, we multiply (20) by an arbitrary smooth (vector-)function $\mathbf{v}$ and integrate by parts over an element κ in the mesh $\mathcal{T}_h$; thereby, we obtain

$$-\int_{\kappa} \mathcal{F}^c(\mathbf{u}) \cdot \nabla \mathbf{v} \, \mathrm{d}\mathbf{x} + \int_{\partial\kappa} \mathcal{F}^c(\mathbf{u}) \cdot \mathbf{n} \, \mathbf{v} \, \mathrm{d}s = 0. \tag{39}$$

To discretize (39), we replace the analytical solution $\mathbf{u}$ by the Galerkin finite element approximation $\mathbf{u}_h$ and the test function $\mathbf{v}$ by $\mathbf{v}_h$, where $\mathbf{u}_h$ and $\mathbf{v}_h$ both belong to the finite element space $\mathbf{V}_{h,p}$. In addition, since the numerical solution $\mathbf{u}_h$ is discontinuous between element interfaces, we must replace the flux $\mathcal{F}^c(\mathbf{u}) \cdot \mathbf{n}$ by a *numerical flux* function $\mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$, which depends on both the interior– and outer–trace of $\mathbf{u}_h$ on $\partial\kappa$, $\kappa \in \mathcal{T}_h$, and the unit outward normal $\mathbf{n}$ to $\partial\kappa$. Thereby, summing over the elements κ in the mesh $\mathcal{T}_h$, yields the DG discretization of (20) as follows: find $\mathbf{u}_h \in \mathbf{V}_{h,p}$ such that

$$-\int_{\Omega} \mathcal{F}^c(\mathbf{u}_h) \cdot \nabla_h \mathbf{v}_h \, \mathrm{d}\mathbf{x} + \sum_{\kappa \in \mathcal{T}_h} \int_{\partial\kappa} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}) \, \mathbf{v}_h^+ \, \mathrm{d}s = 0 \qquad \forall \mathbf{v}_h \in \mathbf{V}_{h,p}. \tag{40}$$

We remark that the replacement of the flux $\mathcal{F}^c(\mathbf{u})\cdot\mathbf{n}$ by the numerical flux function $\mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})$ on the boundary of element κ, κ in $\mathcal{T}_h$, corresponds to the weak imposition of the boundary data. The numerical flux $\mathcal{H}(\cdot,\cdot,\cdot)$ must be consistent and conservative; here, we recall the following definitions

(i) $\mathcal{H}(\cdot,\cdot,\cdot)|_{\partial\kappa}$ is consistent with the flux $\mathcal{F}^c(\cdot)\cdot\mathbf{n}$, if for each κ in $\mathcal{T}_h$ we have that

$$\mathcal{H}(\mathbf{v},\mathbf{v},\mathbf{n})|_{\partial\kappa} = \mathcal{F}^c(\mathbf{v})\cdot\mathbf{n} \quad \forall \kappa \in \mathcal{T}_h;$$

(ii) $\mathcal{H}(\cdot,\cdot,\cdot)$ is conservative, if given any two neighboring elements κ and κ' from the finite element partition $\mathcal{T}_h$, at each point $\mathbf{x} \in \partial\kappa \cap \partial\kappa' \neq \emptyset$, noting that $\mathbf{n}_{\kappa'} = -\mathbf{n}$, we have that

$$\mathcal{H}(\mathbf{v},\mathbf{w},\mathbf{n}) = -\mathcal{H}(\mathbf{w},\mathbf{v},-\mathbf{n}).$$

There are several numerical flux functions satisfying these conditions, such as the Godunov, Engquist–Osher, Lax–Friedrichs, Roe or the Vijayasundaram flux. As examples, here we consider three different numerical fluxes:

- The **(local) Lax–Friedrichs flux** $\mathcal{H}_{LF}(\cdot,\cdot,\cdot)$ is defined by

$$\mathcal{H}_{LF}(\mathbf{u}^+,\mathbf{u}^-,\mathbf{n})|_{\partial\kappa} = \tfrac{1}{2}\left(\mathcal{F}^c(\mathbf{u}^+)\cdot\mathbf{n} + \mathcal{F}^c(\mathbf{u}^-)\cdot\mathbf{n} + \alpha\left(\mathbf{u}^+ - \mathbf{u}^-\right)\right),$$

  for $\kappa \in \mathcal{T}_h$, where α is the maximum over $\mathbf{u}^+$ and $\mathbf{u}^-$,

$$\alpha = \max_{\mathbf{v}=\mathbf{u}^+,\mathbf{u}^-} \{|\lambda(A_n(\mathbf{v},\mathbf{n}))|\},$$

  of the largest eigenvalue (in absolute value) $|\lambda(A_n)|$ of the matrix $A_n(\mathbf{u},\mathbf{n}) = \sum_{i=0}^{3} n_i A_i(\mathbf{u})$ defined in (24).

- The **Vijayasundaram flux** $\mathcal{H}_V(\cdot,\cdot,\cdot)$ is defined by

$$\mathcal{H}_V(\mathbf{u}^+,\mathbf{u}^-,\mathbf{n})|_{\partial\kappa} = A_n^+(\bar{\mathbf{u}},\mathbf{n})\mathbf{u}^+ + A_n^-(\bar{\mathbf{u}},\mathbf{n})\mathbf{u}^- \quad \text{for } \kappa \in \mathcal{T}_h,$$

  where $A_n^+(\bar{\mathbf{u}},\mathbf{n})$ and $A_n^-(\bar{\mathbf{u}},\mathbf{n})$ denote the positive and negative parts, cf. (27), of the matrix $A_n(\bar{\mathbf{u}},\mathbf{n})$, respectively, evaluated at an average state $\bar{\mathbf{u}}$ between $\mathbf{u}^+$ and $\mathbf{u}^-$.

- The **HLLE flux** $\mathcal{H}_{HLLE}(\cdot,\cdot,\cdot)$ is given by

$$\mathcal{H}_{HLLE}(\mathbf{u}^+,\mathbf{u}^-,\mathbf{n})|_{\partial\kappa} = \tfrac{1}{\lambda^+-\lambda^-}\left(\lambda^+\mathcal{F}^c(\mathbf{u}^+)\cdot\mathbf{n} - \lambda^-\mathcal{F}^c(\mathbf{u}^-)\cdot\mathbf{n} - \lambda^+\lambda^-\left(\mathbf{u}^+-\mathbf{u}^-\right)\right),$$

  where $\lambda^+ = \max(\lambda_{\max},0)$ and $\lambda^- = \min(\lambda_{\min},0)$.

Boundary conditions For boundary faces $\partial\kappa\cap\Gamma\neq\emptyset$ we replace $\mathbf{u}_h^-$ by an appropriate boundary function $\mathbf{u}_\Gamma(\mathbf{u}_h^+)$ which realizes the boundary conditions to be imposed.

First we define several *farfield boundary conditions*:

- Supersonic inflow corresponds to Dirichlet boundary conditions where

$$\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{g}_D = \mathbf{u}_\infty \quad \text{on} \quad \Gamma_{\text{D,sup}}.$$

- Supersonic outflow corresponds to Neumann boundary conditions where

$$\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{u} \quad \text{on} \quad \Gamma_{\text{N}}.$$

- The subsonic inflow boundary condition takes the pressure from the flow field and imposes all other variables based on freestream conditions $\mathbf{u}_\infty$, i.e.,

$$\mathbf{u}_\Gamma(\mathbf{u}) = \left(\rho_\infty, \rho_\infty v_{1,\infty}, \rho_\infty v_{2,\infty}, \rho_\infty v_{3,\infty}, \frac{p(\mathbf{u})}{\gamma-1} + \rho_\infty\left(v_{1,\infty}^2+v_{2,\infty}^2+v_{3,\infty}^2\right)\right)^\top \quad \text{on} \quad \Gamma_{\text{D,sub-in}}.$$

  Here, $p\equiv p(\mathbf{u})$ denotes the pressure evaluated using the equation of state (23).

- The subsonic outflow boundary condition imposes an outflow pressure p_{out} and takes all other variables from the flow field, i.e.,

$$\mathbf{u}_\Gamma(\mathbf{u}) = \left(u_1, u_2, u_3, u_4, \frac{p_{\text{out}}}{\gamma-1} + \frac{u_2^2+u_3^2+u_4^2}{2u_1}\right)^\top \quad \text{on} \quad \Gamma_{\text{D,sub-out}}.$$

- The characteristic farfield boundary condition imposes Dirichlet boundary conditions based on free-stream conditions on characteristic inflow variables. No boundary conditions are imposed on characteristic outflow variables. This corresponds to using the Vijayasundaram flux on the farfield boundary.

Finally, we define the following *slip wall boundary condition*:

- For slip wall boundary conditions used at reflective walls we set

$$\mathbf{u}_\Gamma(\mathbf{u}) = \begin{pmatrix} 1 & 0 & 0 & 0 & \\ 0 & 1-n_1^2 & -n_1n_2 & -n_1n_3 & 0 \\ 0 & -n_1n_2 & 1-n_2^2 & -n_2n_3 & 0 \\ 0 & -n_1n_3 & -n_2n_3 & 1-n_3^2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{u} \quad \text{on} \quad \Gamma_W, \tag{41}$$

  which originates from $\mathbf{u}$ by subtracting the normal velocity component of $\mathbf{u}$, i.e., $\mathbf{v} = (v_1, v_2, v_3)$ is replaced by $\mathbf{v}_\Gamma = \mathbf{v} - (\mathbf{v} \cdot \mathbf{n})\mathbf{n}$ which ensures that the normal velocity component vanishes, $\mathbf{v}_\Gamma \cdot \mathbf{n} = 0$.

Given the boundary function $\mathbf{u}_\Gamma(\mathbf{u}_h^+)$ as defined above the DG discretization of (20) including boundary conditions is given as follows: find $\mathbf{u}_h \in \mathbf{V}_{h,p}$ such that

$$\begin{aligned} \mathcal{N}(\mathbf{u}_h, \mathbf{v}_h) \equiv -\int_\Omega \mathcal{F}^c(\mathbf{u}_h) \cdot \nabla_h \mathbf{v}_h \,\mathrm{d}\mathbf{x} + \sum_{\kappa \in \mathcal{T}_h} \int_{\partial\kappa \setminus \Gamma} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n})\, \mathbf{v}_h^+ \,\mathrm{d}s \\ + \int_\Gamma \mathcal{H}_\Gamma(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n})\, \mathbf{v}_h^+ \,\mathrm{d}s = 0 \quad \forall \mathbf{v}_h \in \mathbf{V}_{h,p}, \end{aligned} \tag{42}$$

where $\mathcal{H}_\Gamma$ is usually the same numerical flux $\mathcal{H}$ as used on interior faces $\partial\kappa \setminus \Gamma$, $\kappa \in \mathcal{T}_h$. However, in order to ensure adjoint consistency, see e.g. [54, 56], the numerical flux $\mathcal{H}_\Gamma$ at slip wall boundaries is given by

$$\mathcal{H}_\Gamma(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}) = \mathbf{n} \cdot \mathcal{F}_\Gamma^c(\mathbf{u}_h^+) = \mathbf{n} \cdot \mathcal{F}^c(\mathbf{u}_\Gamma(\mathbf{u}_h^+)) \qquad \text{on } \Gamma_W. \tag{43}$$

3.5 Consistency and adjoint consistency

Using integration by parts on (42) we obtain the primal residual form: find $\mathbf{u}_h \in \mathbf{V}_{h,p}$ such that

$$\mathcal{R}(\mathbf{u}_h, \mathbf{v}_h) \equiv \int_\Omega \mathbf{R}(\mathbf{u}_h) \cdot \mathbf{v}_h \,\mathrm{d}\mathbf{x} + \sum_{\kappa \in \mathcal{T}_h} \int_{\partial\kappa \setminus \Gamma} \mathbf{r}(\mathbf{u}_h) \cdot \mathbf{v}_h^+ \,\mathrm{d}s + \int_\Gamma \mathbf{r}_\Gamma(\mathbf{u}_h) \cdot \mathbf{v}_h^+ \,\mathrm{d}s = 0 \tag{44}$$

for all $\mathbf{v}_h \in \mathbf{V}_{h,p}$, where the primal residuals are given by

$$\begin{aligned} \mathbf{R}(\mathbf{u}_h) &= -\nabla \cdot \mathcal{F}^c(\mathbf{u}_h) && \text{in } \kappa,\ \kappa \in \mathcal{T}_h, \\ \mathbf{r}(\mathbf{u}_h) &= \mathbf{n} \cdot \mathcal{F}^c(\mathbf{u}_h^+) - \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}^+) && \text{on } \partial\kappa \setminus \Gamma, \kappa \in \mathcal{T}_h, \\ \mathbf{r}_\Gamma(\mathbf{u}_h) &= \mathbf{n} \cdot \mathcal{F}^c(\mathbf{u}_h^+) - \mathcal{H}_\Gamma(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}^+) && \text{on } \Gamma. \end{aligned} \tag{45}$$

Given the consistency of the numerical flux, $\mathcal{H}(\mathbf{w}, \mathbf{w}, \mathbf{n}) = \mathbf{n} \cdot \mathcal{F}^c(\mathbf{w})$, and the consistency of the boundary function, i.e., $\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{u}$ for the analytical solution $\mathbf{u}$ to (20), we find that $\mathbf{u}$ satisfies the following equations

$$\mathbf{R}(\mathbf{u}) = 0 \ \text{ in } \kappa, \kappa \in \mathcal{T}_h, \qquad \mathbf{r}(\mathbf{u}) = 0 \ \text{ on } \partial\kappa \setminus \Gamma, \kappa \in \mathcal{T}_h, \qquad \mathbf{r}_\Gamma(\mathbf{u}) = 0 \ \text{ on } \Gamma. \tag{46}$$

Thereby, (42) is a *consistent* discretization of (20), i.e., the analytical solution $\mathbf{u} \in \mathbf{V}$ to the primal problem (20) satisfies the equation

$$\mathcal{N}(\mathbf{u}, \mathbf{v}) = 0 \quad \forall \mathbf{v} \in \mathbf{V}, \tag{47}$$

where $\mathbf{V}$ is some suitably chosen function space including the analytical solution $\mathbf{u} \in \mathbf{V}$ to the primal problem (20) and satisfying $\mathbf{V}_{h,p} \subset \mathbf{V}$; see [7, 54] for the choice of $\mathbf{V}$ in the case of DG methods.

Furthermore, we note that the discretization (42) is *adjoint consistent* in combination with the modified target quantity $\hat{J}(\mathbf{u}_h) = J(\mathbf{u}_\Gamma(\mathbf{u}_h))$, where $J(\cdot)$ is an aerodynamic force coefficient as given in (33). In fact, denoting by $\mathcal{N}'[\mathbf{u}_h](\mathbf{w}_h, \mathbf{v}_h)$ the Fréchet derivative of $\mathcal{N}(\mathbf{u}_h, \mathbf{v}_h)$ and by $\hat{J}'[\mathbf{u}_h](\mathbf{w}_h)$ the Fréchet derivative of $\hat{J}(\mathbf{u}_h)$, both with respect to $\mathbf{u}_h$ in the direction of $\mathbf{w}_h \in \mathbf{V}_{h,p}$, then the discrete adjoint problem given by: find $\mathbf{z}_h \in \mathbf{V}_{h,p}$ such that

$$\mathcal{N}'[\mathbf{u}_h](\mathbf{w}_h, \mathbf{z}_h) = \hat{J}'[\mathbf{u}_h](\mathbf{w}_h) \quad \forall \mathbf{w}_h \in \mathbf{V}_{h,p},$$

is a consistent discretization of the continuous adjoint problem (36) with adjoint boundary conditions (37), see e.g. [54, 56]. This means, that the analytical solution $\mathbf{z}$ to (36),(37) satisfies

$$\mathcal{N}'[\mathbf{u}](\mathbf{w}, \mathbf{z}) = \hat{J}'[\mathbf{u}](\mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V}. \tag{48}$$

Finally, noting that $\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{u}$ holds for the analytical solution $\mathbf{u}$, we see that the modification $\hat{J}(\mathbf{u}_h)$ of the target quantity $J(\mathbf{u}_h)$ is *consistent*, i.e., their values coincide, $\hat{J}(\mathbf{u}) = J(\mathbf{u})$, if evaluated for the analytical solution $\mathbf{u}$. Finally, we note that we will omit the $\hat{}$ notation and write J instead of $\hat{J}$ in the following.

3.6 The compressible Navier-Stokes equations

The compressible Euler equations as considered in Section 3.1 serve as a simple model for gas flows. In fact, while ignoring all viscous effects they describe an inviscid compressible flow. In the following, we will enrich the physical model by including also viscous terms. The resulting compressible Navier-Stokes equations serve as a model for laminar viscous compressible flows.

As in Section 3.1, the variables ρ, $\mathbf{v} = (v_1, v_2, v_3)^\top$, p and E denote the density, velocity vector, pressure and specific total energy, respectively. Furthermore, T denotes the temperature. The equations of motion are given by

$$\nabla \cdot (\mathcal{F}^c(\mathbf{u}) - \mathcal{F}^v(\mathbf{u}, \nabla\mathbf{u})) \equiv \frac{\partial}{\partial x_k}\mathbf{f}_k^c(\mathbf{u}) - \frac{\partial}{\partial x_k}\mathbf{f}_k^v(\mathbf{u}, \nabla\mathbf{u}) = 0 \quad \text{in } \Omega. \tag{49}$$

The vector of conservative variables $\mathbf{u}$ is given by $\mathbf{u} = (\rho, \rho v_1, \rho v_2, \rho v_3, \rho E)^\top$ and the convective fluxes $\mathbf{f}_k^c$, $k = 1, 2, 3$, are given by (21). Furthermore, the viscous fluxes $\mathbf{f}_k^v$, $k = 1, 2, 3$, are defined by

$$\mathbf{f}_k^v(\mathbf{u}, \nabla\mathbf{u}) = \begin{pmatrix} 0 \\ \tau_{1k} \\ \tau_{2k} \\ \tau_{3k} \\ \tau_{kl}v_l + \mathcal{K}T_{x_k} \end{pmatrix}, \quad k = 1, 2, 3,$$

where $\mathcal{K}$ is the thermal conductivity coefficient. Finally, the viscous stress tensor is defined by

$$\tau = \mu \left(\nabla\mathbf{v} + (\nabla\mathbf{v})^\top - \tfrac{2}{3}(\nabla \cdot \mathbf{v})I \right),$$

where μ is the dynamic viscosity coefficient, and the temperature T is given by $e = c_v T$; thus

$$\mathcal{K}T = \tfrac{\mu\gamma}{Pr}\left(E - \tfrac{1}{2}\mathbf{v}^2\right),$$

where $Pr = 0.72$ is the Prandtl number.

For the purposes of discretization, we rewrite the compressible Navier–Stokes equations (49) in the following (equivalent) form:

$$\nabla\cdot(\mathcal{F}^c(\mathbf{u}) - G(\mathbf{u})\nabla\mathbf{u}) \equiv \frac{\partial}{\partial x_k}\left(\mathbf{f}_k^c(\mathbf{u}) - G_{kl}(\mathbf{u})\frac{\partial \mathbf{u}}{\partial x_l}\right) = 0 \quad \text{in } \Omega.$$

Here, the matrices $G_{kl}(\mathbf{u}) = \partial \mathbf{f}_k^v(\mathbf{u},\nabla\mathbf{u})/\partial u_{x_l}$, for $k,l = 1,2,3$, are the homogeneity tensors defined by $\mathbf{f}_k^v(\mathbf{u},\nabla\mathbf{u}) = G_{kl}(\mathbf{u})\partial\mathbf{u}/\partial x_l$, $k = 1,2,3$.

As for the compressible Euler equations we consider supersonic and subsonic inflow and outflow boundary conditions. Furthermore, we distinguish between *isothermal* and *adiabatic* wall boundary conditions. To this end, decomposing $\Gamma_W = \Gamma_{\text{iso}} \cup \Gamma_{\text{adia}}$, we set

$$\mathbf{v} = \mathbf{0} \ \text{ on } \ \Gamma_W, \qquad T = T_{\text{wall}} \ \text{ on } \ \Gamma_{\text{iso}}, \qquad \mathbf{n}\cdot\nabla T = 0 \ \text{ on } \ \Gamma_{\text{adia}}, \tag{50}$$

where T_{wall} is a given wall temperature.

3.7 The adjoint equations to the compressible Navier-Stokes equations

The most important target quantities in viscous compressible flows are the total (i.e., the pressure induced plus viscous) drag and lift coefficients, C_{d} and C_{l}, defined by

$$J(\mathbf{u}) = \int_\Gamma j(\mathbf{u})\,\mathrm{d}s = \frac{1}{C_\infty}\int_{\Gamma_W}(p\,\mathbf{n} - \underline{\tau}\,\mathbf{n})\cdot\boldsymbol{\psi}\,\mathrm{d}s = \frac{1}{C_\infty}\int_{\Gamma_W}(p\,n_i - \tau_{ij}n_j)\,\psi_i\,\mathrm{d}s, \tag{51}$$

where C_∞ and $\boldsymbol{\psi}$ are as in (33). In order to derive the adjoint problem, we multiply the left hand side of (49) by $\mathbf{z}$, integrate by parts and linearize about $\mathbf{u}$ to obtain

$$\begin{aligned}&(\nabla\cdot(\mathcal{F}^c_{\mathbf{u}}\mathbf{w} - \mathcal{F}^v_{\mathbf{u}}\mathbf{w} - \mathcal{F}^v_{\nabla\mathbf{u}}\nabla\mathbf{w}),\mathbf{z})\\ &\qquad = -\left((\mathcal{F}^c_{\mathbf{u}} - \mathcal{F}^v_{\mathbf{u}})\,\mathbf{w} - \mathcal{F}^v_{\nabla\mathbf{u}}\nabla\mathbf{w},\nabla\mathbf{z}\right) + (\mathbf{n}\cdot(\mathcal{F}^c_{\mathbf{u}}\mathbf{w} - \mathcal{F}^v_{\mathbf{u}}\mathbf{w} - \mathcal{F}^v_{\nabla\mathbf{u}}\nabla\mathbf{w}),\mathbf{z})_\Gamma,\end{aligned}$$

where $\mathcal{F}^v_{\mathbf{u}} := \partial_{\mathbf{u}}\mathcal{F}^v(\mathbf{u},\nabla\mathbf{u}) = G'[\mathbf{u}]\nabla\mathbf{u}$ and $\mathcal{F}^v_{\nabla\mathbf{u}} := \partial_{\nabla\mathbf{u}}\mathcal{F}^v(\mathbf{u},\nabla\mathbf{u}) = G(\mathbf{u})$ denote the derivatives of $\mathcal{F}^v$ with respect to $\mathbf{u}$ and $\nabla\mathbf{u}$, respectively. Using integration by parts once more, we obtain the following variational formulation of the continuous adjoint problem: find $\mathbf{z}$ such that

$$\begin{aligned}&-\left(\mathbf{w},(\mathcal{F}^c_{\mathbf{u}} - \mathcal{F}^v_{\mathbf{u}})^\top\nabla\mathbf{z}\right) - \left(\mathbf{w},\nabla\cdot\left((\mathcal{F}^v_{\nabla\mathbf{u}})^\top\nabla\mathbf{z}\right)\right) + \left(\mathbf{w},\mathbf{n}\cdot\left((\mathcal{F}^v_{\nabla\mathbf{u}})^\top\nabla\mathbf{z}\right)\right)_\Gamma\\ &\qquad + \left(\mathbf{w},(\mathbf{n}\cdot(\mathcal{F}^c_{\mathbf{u}} - \mathcal{F}^v_{\mathbf{u}}))^\top\mathbf{z}\right)_\Gamma - \left(\nabla\mathbf{w},(\mathbf{n}\cdot\mathcal{F}^v_{\nabla\mathbf{u}})^\top\mathbf{z}\right)_\Gamma = J'[\mathbf{u}](\mathbf{w}) \quad \forall\mathbf{w}\in\mathbf{V}.\end{aligned}$$

Given that

$$\begin{aligned}J'[\mathbf{u}](\mathbf{w}) =& \frac{1}{C_\infty}\int_{\Gamma_W}(p_{\mathbf{u}}[\mathbf{u}]\,\mathbf{n} - \underline{\tau}_{\mathbf{u}}[\mathbf{u}]\,\mathbf{n})\cdot\boldsymbol{\psi}\,\mathbf{w} - (\underline{\tau}_{\nabla\mathbf{u}}[\mathbf{u}]\,\mathbf{n})\cdot\boldsymbol{\psi}\,\nabla\mathbf{w}\,\mathrm{d}s\\ =& \left(\mathbf{w},\tfrac{1}{C_\infty}(p_{\mathbf{u}}\,\mathbf{n} - \underline{\tau}_{\mathbf{u}}\,\mathbf{n})\cdot\boldsymbol{\psi}\right)_{\Gamma_W} - \left(\nabla\mathbf{w},\tfrac{1}{C_\infty}(\underline{\tau}_{\nabla\mathbf{u}}\,\mathbf{n})\cdot\boldsymbol{\psi}\right)_{\Gamma_W},\end{aligned} \tag{52}$$

we see that the adjoint solution $\mathbf{z}$ satisfies the following equation

$$-\left(\mathcal{F}^c_{\mathbf{u}} - \mathcal{F}^v_{\mathbf{u}}\right)^\top \nabla \mathbf{z} - \nabla \cdot \left(\left(\mathcal{F}^v_{\nabla \mathbf{u}}\right)^\top \nabla \mathbf{z}\right) = 0, \tag{53}$$

subject to the boundary conditions on $\Gamma_W = \Gamma_{\text{iso}} \cup \Gamma_{\text{adia}}$,

$$\left(\mathbf{n} \cdot \left(\mathcal{F}^c_{\mathbf{u}} - \mathcal{F}^v_{\mathbf{u}}\right)\right)^\top \mathbf{z} + \mathbf{n} \cdot \left(\left(\mathcal{F}^v_{\nabla \mathbf{u}}\right)^\top \nabla \mathbf{z}\right) = \tfrac{1}{C_\infty} \left(p_{\mathbf{u}}\, \mathbf{n} - \underline{\tau}_{\mathbf{u}}\, \mathbf{n}\right) \cdot \boldsymbol{\psi}, \tag{54}$$

$$\left(\mathbf{n} \cdot \mathcal{F}^v_{\nabla \mathbf{u}}\right)^\top \mathbf{z} = \tfrac{1}{C_\infty} \left(\underline{\tau}_{\nabla \mathbf{u}}\, \mathbf{n}\right) \cdot \boldsymbol{\psi}. \tag{55}$$

At wall boundaries Γ_W, where $\mathbf{v} = (v_1, v_2, v_3)^\top = 0$, the normal viscous flux reduces to $\mathbf{n} \cdot \mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u}) = (0, (\tau \mathbf{n})_1, (\tau \mathbf{n})_2, (\tau \mathbf{n})_3, \mathcal{K} \mathbf{n} \cdot \nabla T)^\top$. Hence, (55) is fulfilled provided $\mathbf{z}$ satisfies

$$\begin{pmatrix} 0 \\ (\tau_{\nabla \mathbf{u}} \mathbf{n})_1\, z_2 \\ (\tau_{\nabla \mathbf{u}} \mathbf{n})_2\, z_3 \\ (\tau_{\nabla \mathbf{u}} \mathbf{n})_3\, z_4 \\ \mathcal{K} \mathbf{n} \cdot \nabla T_{\nabla \mathbf{u}}\, z_5 \end{pmatrix} = \frac{1}{C_\infty} \begin{pmatrix} 0 \\ (\tau_{\nabla \mathbf{u}} \mathbf{n})_1\, \psi_1 \\ (\tau_{\nabla \mathbf{u}} \mathbf{n})_2\, \psi_2 \\ (\tau_{\nabla \mathbf{u}} \mathbf{n})_3\, \psi_3 \\ 0 \end{pmatrix}, \tag{56}$$

which reduces to the conditions $z_{i+1} = \frac{1}{C_\infty} \psi_i$, $i = 1, 2, 3$, on Γ_W, and $z_5 = 0$ on Γ_{iso}. At adiabatic boundaries we have $\mathbf{n} \cdot \nabla T = 0$ and the last condition in (56) vanishes. Substituting into (54) we obtain $\mathbf{n} \cdot ((\mathcal{F}^v_{\nabla \mathbf{u}})^\top \nabla \mathbf{z}) = 0$ on Γ_W which at adiabatic boundaries reduces to $\mathbf{n} \cdot \nabla z_5 = 0$. On isothermal boundaries no additional boundary condition is obtained. In summary, the boundary conditions of the adjoint problem (53) to the compressible Navier-Stokes equations are given by

$$z_{i+1} = \tfrac{1}{C_\infty} \psi_i, \ i = 1, 2, 3, \quad \text{on } \Gamma_W, \qquad z_5 = 0 \ \text{ on } \Gamma_{\text{iso}}, \qquad \mathbf{n} \cdot \nabla z_5 = 0 \ \text{ on } \Gamma_{\text{adia}}. \tag{57}$$

3.8 DG discretization of the compressible Navier-Stokes equations

In addition to the vector-valued discrete function space $\mathbf{V}_{h,p}$ defined in Section 3.4 we now introduce the tensor-valued discrete function space $\underline{\Sigma}_{h,p}$ consisting of tensor-valued polynomial functions of degree $p \geq 0$, defined by

$$\begin{aligned} \underline{\Sigma}_{h,p} = \{\underline{\tau} \in [L_2(\Omega)]^{5\times 3} : & \underline{\tau}|_\kappa \circ F_\kappa \in [Q_p(\hat{\kappa})]^{5\times 3} \ \text{ if } \hat{\kappa} \text{ is the unit hypercube, and} \\ & \underline{\tau}|_\kappa \circ F_\kappa \in [P_p(\hat{\kappa})]^{5\times 3} \ \text{ if } \hat{\kappa} \text{ is the unit simplex}, \kappa \in \mathcal{T}_h\}. \end{aligned}$$

An *interior face* of $\mathcal{T}_h$ is defined as the (non-empty) two–dimensional interior of $\partial\kappa^+ \cap \partial\kappa^-$, where κ^+ and κ^- are two adjacent elements of $\mathcal{T}_h$, not necessarily matching. A *boundary face* of $\mathcal{T}_h$ is defined as the (non-empty) two–dimensional interior of $\partial\kappa \cap \Gamma$, where κ is a boundary element of $\mathcal{T}_h$. We denote by $\Gamma_{\mathcal{I}}$ the union of all interior faces of $\mathcal{T}_h$. Furthermore, we define some jump and mean value operators for vector- and matrix-valued functions. To this end, let κ^+ and κ^- be two adjacent elements of $\mathcal{T}_h$ and $\mathbf{x}$ be an arbitrary point on the interior face $f = \partial\kappa^+ \cap \partial\kappa^- \subset \Gamma_{\mathcal{I}}$. Moreover, let $\mathbf{v}$ and $\underline{\tau}$ be vector- and matrix-valued functions, respectively, that are smooth inside each element $\kappa^\pm$. By $\mathbf{v}^\pm := \mathbf{v}|_{\partial\kappa^\pm}$ and $\underline{\tau}^\pm := \underline{\tau}|_{\partial\kappa^\pm}$ we denote the traces of, respectively, $\mathbf{v}$ and $\underline{\tau}$ on f taken from within the interior of $\kappa^\pm$, respectively. Then, we define the averages at $\mathbf{x} \in f$ by $\{\!\{\mathbf{v}\}\!\} = (\mathbf{v}^+ + \mathbf{v}^-)/2$ and $\{\!\{\underline{\tau}\}\!\} = (\underline{\tau}^+ + \underline{\tau}^-)/2$. Similarly, the jump at $\mathbf{x} \in f$ is given by $\underline{[\![\mathbf{v}]\!]} = \mathbf{v}^+ \otimes \mathbf{n}_{\kappa^+} + \mathbf{v}^- \otimes \mathbf{n}_{\kappa^-}$.

On a boundary face $f \subset \Gamma$, we set $\{\!\{\mathbf{v}\}\!\} = \mathbf{v}$, $\{\!\{\underline{\tau}\}\!\} = \underline{\tau}$ and $\underline{[\![\mathbf{v}]\!]} = \mathbf{v} \otimes \mathbf{n}$. For matrices $\underline{\sigma}, \underline{\tau} \in \mathbb{R}^{m\times n}$, $m, n \geq 1$, we use the standard notation $\underline{\sigma} : \underline{\tau} = \sum_{k=1}^{m}\sum_{l=1}^{n} \sigma_{kl}\tau_{kl}$; additionally, for vectors $\mathbf{v} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^n$, the matrix $\mathbf{v} \otimes \mathbf{w} \in \mathbb{R}^{m\times n}$ is defined by $(\mathbf{v} \otimes \mathbf{w})_{kl} = v_k\, w_l$.

The DG discretization of the three–dimensional compressible Navier-Stokes equations (49) is given by: find $\mathbf{u}_h \in \mathbf{V}_{h,p}$ such that

$$\begin{aligned}
\mathcal{N}(\mathbf{u}_h, \mathbf{v}) \;\equiv\; & -\int_\Omega \mathcal{F}^c(\mathbf{u}_h) : \nabla_h \mathbf{v}\,\mathrm{d}\mathbf{x} + \sum_{\kappa\in\mathcal{T}_h}\int_{\partial\kappa\setminus\Gamma} \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}^+)\cdot\mathbf{v}^+\,\mathrm{d}s \\
& + \int_\Omega \mathcal{F}^v(\mathbf{u}_h, \nabla_h\mathbf{u}_h) : \nabla_h\mathbf{v}\,\mathrm{d}\mathbf{x} - \int_{\Gamma_\mathcal{I}} \{\!\{\mathcal{F}^v(\mathbf{u}_h, \nabla_h\mathbf{u}_h)\}\!\} : \underline{[\![\mathbf{v}]\!]}\,\mathrm{d}s \\
& - \int_{\Gamma_\mathcal{I}} \{\!\{G^\top(\mathbf{u}_h)\nabla_h\mathbf{v}\}\!\} : \underline{[\![\mathbf{u}_h]\!]}\,\mathrm{d}s + \int_{\Gamma_\mathcal{I}} \underline{\delta}(\mathbf{u}_h) : \underline{[\![\mathbf{v}]\!]}\,\mathrm{d}s \\
& + \mathcal{N}_\Gamma(\mathbf{u}_h, \mathbf{v}) = 0
\end{aligned} \tag{58}$$

for all $\mathbf{v}$ in $\mathbf{V}_{h,p}$. Here, the numerical flux $\mathcal{H}$ is as described in Section 3.4. For the penalization term we consider the interior penalty (IP) scheme, [63], and the second scheme of Bassi and Rebay (BR2), [16, 17]:

$$\begin{aligned}
&\underline{\delta}(\mathbf{u}_h) = \underline{\delta}^{\mathrm{IP}}(\mathbf{u}_h) = C_{\mathrm{IP}}\frac{p^2}{h_f}\{\!\{G(\mathbf{u}_h)\}\!\}\underline{[\![\mathbf{u}_h]\!]} && \text{for IP [63]}, \\
&\underline{\delta}(\mathbf{u}_h) = \underline{\delta}^{\mathrm{BR2}}(\mathbf{u}_h) = C_{\mathrm{BR2}}\{\!\{\underline{L}_0^e(\mathbf{u}_h)\}\!\} && \text{for BR2 [16, 17]}.
\end{aligned} \tag{59}$$

Here, h_f represents the element dimension orthogonal to the face $f \subset \partial\kappa^+ \cap \partial\kappa^-$, where κ^+ and κ^- are adjacent elements, cf. Section 5.4. C_{IP} is a positive constant, which, for reasons of stability, must be chosen sufficiently large. Furthermore, the local lifting operator $\mathbf{L}_0^f(\mathbf{u}_h) \in \underline{\Sigma}_{h,p}$ is defined by:

$$\int_{\Omega_e} \mathbf{L}_0^f(\mathbf{u}_h) : \underline{\tau}\,\mathrm{d}\mathbf{x} = \int_e \underline{[\![\mathbf{u}_h]\!]} : \{\!\{G^\top(\mathbf{u}_h)\underline{\tau}\}\!\}\,\mathrm{d}s \qquad \forall \underline{\tau} \in \underline{\Sigma}_{h,p},$$

where $\Omega_e = \kappa_e^+ \cup \kappa_e^-$ with $e = \partial\kappa_e^+ \cap \partial\kappa_e^-$.

Finally, the boundary terms included in $\mathcal{N}_\Gamma(\mathbf{u}_h, \mathbf{v})$ are given by

$$\begin{aligned}
\mathcal{N}_\Gamma(\mathbf{u}_h, \mathbf{v}) = & \int_\Gamma \mathcal{H}_\Gamma(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}^+)\cdot\mathbf{v}^+\,\mathrm{d}s + \int_\Gamma \underline{\delta}_\Gamma(\mathbf{u}_h^+) : \mathbf{v}\otimes\mathbf{n}\,\mathrm{d}s, \\
& - \int_\Gamma \mathbf{n}\cdot\mathcal{F}_\Gamma^v(\mathbf{u}_h^+, \nabla_h\mathbf{u}_h^+)\,\mathbf{v}^+\,\mathrm{d}s \\
& - \int_\Gamma \left(G_\Gamma^\top(\mathbf{u}_h^+)\nabla_h\mathbf{v}_h^+\right) : \left(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)\right)\otimes\mathbf{n}\,\mathrm{d}s,
\end{aligned} \tag{60}$$

where the penalization term on the boundary is given by

$$\begin{aligned}
&\underline{\delta}_\Gamma^{\mathrm{IP}}(\mathbf{u}_h) = C_{\mathrm{IP}}\frac{p^2}{h_e}G_\Gamma(\mathbf{u}_h)\left(\mathbf{u}_h - \mathbf{u}_\Gamma(\mathbf{u}_h)\right)\otimes\mathbf{n} && \text{for IP [63]}, \\
&\underline{\delta}_\Gamma^{\mathrm{BR2}}(\mathbf{u}_h) = C_{\mathrm{BR2}}\underline{L}_\Gamma^e(\mathbf{u}_h) && \text{for BR2 [16, 17]}.
\end{aligned} \tag{61}$$

Here, the local lifting operator $\mathbf{L}_\Gamma^f(\mathbf{u}_h) \in \underline{\Sigma}_{h,p}$ on Γ is defined by:

$$\int_\kappa \mathbf{L}_\Gamma^f(\mathbf{u}_h) : \underline{\tau}\,\mathrm{d}\mathbf{x} = \int_e \left(\mathbf{u}_h - \mathbf{u}_\Gamma(\mathbf{u}_h)\right)\otimes\mathbf{n} : \left(G_\Gamma^\top(\mathbf{u}_h)\underline{\tau}\right)\,\mathrm{d}s \qquad \forall \underline{\tau} \in \underline{\Sigma}_{h,p}$$

for all $\kappa \in \mathcal{T}_h$, such that $\partial\kappa \cap \Gamma = e$. Furthermore, the viscous boundary flux $\mathcal{F}^v_\Gamma$ and the corresponding homogeneity tensor G_Γ are defined by

$$\mathcal{F}^v_\Gamma(\mathbf{u}_h, \nabla\mathbf{u}_h) = \mathcal{F}^v(\mathbf{u}_\Gamma(\mathbf{u}_h), \nabla\mathbf{u}_h) = G_\Gamma(\mathbf{u}_h)\nabla\mathbf{u}_h = G(\mathbf{u}_\Gamma(\mathbf{u}_h))\nabla\mathbf{u}_h. \tag{62}$$

Furthermore, on adiabatic boundaries $\Gamma_{\text{adia}} \subset \Gamma_W$, $\mathcal{F}^v_\Gamma$ and G_Γ are modified such that $\mathbf{n}\cdot\nabla T = 0$. Finally, as in (43) we define

$$\mathcal{H}_\Gamma(\mathbf{u}_h^+, \mathbf{u}_\Gamma(\mathbf{u}_h^+), \mathbf{n}) = \mathbf{n}\cdot\mathcal{F}^c_\Gamma(\mathbf{u}_h^+) = \mathbf{n}\cdot\mathcal{F}^c(\mathbf{u}_\Gamma(\mathbf{u}_h^+)), \tag{63}$$

where the boundary function $\mathbf{u}_\Gamma(\cdot)$ is given by $\mathbf{u}_\Gamma(\mathbf{w}) = (w_1, 0, 0, 0, w_5)^\top$ on Γ_{adia}, and by $\mathbf{u}_\Gamma(\mathbf{w}) = (w_1, 0, 0, 0, w_1 c_v T_{\text{wall}})^\top$ on Γ_{iso}, see Section 3.4 for the treatment of other boundary conditions. Finally, we note that the boundary function $\mathbf{u}_\Gamma(\cdot)$ is consistent, i.e., on all boundary parts, $\mathbf{u}_\Gamma(\cdot)$ is chosen such that the analytical solution $\mathbf{u}$ to (49) satisfies $\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{u}$. As a consequence also $\underline{\delta}_\Gamma(\cdot)$ as defined in (61) is consistent. In fact, the analytical solution $\mathbf{u}$ to (49) satisfies $\underline{\delta}_\Gamma(\mathbf{u}) = 0$.

3.9 Consistency and adjoint consistency

Using integration by parts in (58) we obtain the primal residual form: find $\mathbf{u}_h \in \mathbf{V}_{h,p}$ such that

$$\begin{aligned}\mathcal{R}(\mathbf{u}_h, \mathbf{v}_h) \equiv \int_\Omega \mathbf{R}(\mathbf{u}_h)\cdot\mathbf{v}_h \,\mathrm{d}\mathbf{x} + \sum_{\kappa\in\mathcal{T}_h}\int_{\partial\kappa\setminus\Gamma} \mathbf{r}(\mathbf{u}_h)\cdot\mathbf{v}_h^+ + \boldsymbol{\rho}(\mathbf{u}_h) : \nabla\mathbf{v}_h^+ \,\mathrm{d}s \\ + \int_\Gamma \mathbf{r}_\Gamma(\mathbf{u}_h)\cdot\mathbf{v}_h^+ + \boldsymbol{\rho}_\Gamma(\mathbf{u}_h) : \nabla\mathbf{v}_h^+ \,\mathrm{d}s = 0 \quad \forall\mathbf{v}_h \in \mathbf{V}_{h,p},\end{aligned} \tag{64}$$

where the primal residuals are given by

$$\begin{aligned}
\mathbf{R}(\mathbf{u}_h) =& -\nabla\cdot\mathcal{F}^c(\mathbf{u}_h) + \nabla\cdot\mathcal{F}^v(\mathbf{u}_h, \nabla_h\mathbf{u}_h) && \text{in } \kappa, \kappa\in\mathcal{T}_h,\\
\mathbf{r}(\mathbf{u}_h) =& \mathbf{n}\cdot\mathcal{F}^c(\mathbf{u}_h^+) - \mathcal{H}(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}^+) - \frac{1}{2}[\![\mathcal{F}^v(\mathbf{u}_h, \nabla_h\mathbf{u}_h)]\!] - \boldsymbol{\delta}(\mathbf{u}_h)\mathbf{n},\\
\boldsymbol{\rho}(\mathbf{u}_h) =& \frac{1}{2}\left(G(\mathbf{u}_h)\underline{[\![\mathbf{u}_h]\!]}\right)^\top && \text{on } \partial\kappa\setminus\Gamma, \kappa\in\mathcal{T}_h,\\
\mathbf{r}_\Gamma(\mathbf{u}_h) =& \mathbf{n}\cdot\left(\mathcal{F}^c(\mathbf{u}_h^+) - \mathcal{F}^c_\Gamma(\mathbf{u}_h^+) - \mathcal{F}^v(\mathbf{u}_h^+, \nabla\mathbf{u}_h^+) + \mathcal{F}^v_\Gamma(\mathbf{u}_h^+, \nabla\mathbf{u}_h^+)\right) - \boldsymbol{\delta}_\Gamma(\mathbf{u}_h^+)\mathbf{n},\\
\boldsymbol{\rho}_\Gamma(\mathbf{u}_h) =& \left(G_\Gamma^\top(\mathbf{u}_h^+) : \left(\mathbf{u}_h^+ - \mathbf{u}_\Gamma(\mathbf{u}_h^+)\right)\otimes\mathbf{n}\right)^\top && \text{on } \Gamma,
\end{aligned} \tag{65}$$

see [54, 56] for more details. We see that the analytical solution $\mathbf{u}$ to (49) satisfies

$$\mathbf{R}(\mathbf{u}) = 0, \qquad \mathbf{r}(\mathbf{u}) = 0, \qquad \boldsymbol{\rho}(\mathbf{u}) = 0, \qquad \mathbf{r}_\Gamma(\mathbf{u}) = 0, \qquad \boldsymbol{\rho}_\Gamma(\mathbf{u}) = 0,$$

where we used consistency of the numerical flux, $\mathcal{H}(\mathbf{w}, \mathbf{w}, \mathbf{n}) = \mathbf{n}\cdot\mathcal{F}^c(\mathbf{w})$, continuity of $\mathbf{u}$, and the consistency of the boundary function, i.e., $\mathbf{u}$ satisfies $\mathbf{u}_\Gamma(\mathbf{u}) = \mathbf{u}$ on Γ. We conclude that the discretization given in (58) is *consistent*.

Furthermore, we note that the discretization (58) is *adjoint consistent* in combination with the modified target quantity

$$\hat{J}(\mathbf{u}_h) = J(\mathbf{u}_\Gamma(\mathbf{u}_h)) + \int_{\Gamma_W} \underline{\delta}_\Gamma(\mathbf{u}_h) : \mathbf{z}_\Gamma\otimes\mathbf{n}\,\mathrm{d}s, \tag{66}$$

see [54, 56] for more details. Here, $J(\cdot)$ represents the aerodynamic force coefficient defined in (51). Note that the modification $\hat{J}(\mathbf{u}_h)$ of the target quantity $J(\mathbf{u}_h)$ is *consistent*, i.e., their values coincide, $\hat{J}(\mathbf{u}) = J(\mathbf{u})$, if evaluated for the analytical solution $\mathbf{u}$. Again, we note that we will omit the $\hat{}$ notation and write J instead of $\hat{J}$ in the following.

4 Adjoint-based error estimation and adaptive mesh refinement

Important quantities in aerodynamic flow simulations are the aerodynamic force coefficients like the pressure induced as well as the viscous stress induced drag, lift and moment coefficients. In addition to the exact approximation of these quantities, it is of increasing importance, in particular in the field of uncertainty quantification, to estimate the error in the computed values.

While local mesh refinement is required for obtaining reasonably accurate results in applications, the goal of the adaptive refinement is either to compute the force coefficients as accurately as possible within given computing resources or to compute these quantities up to a given tolerance with the minimum computing resources required. In both cases a goal-oriented refinement is needed, i.e., an adaptive refinement strategy specifically targeted to the efficient computation of the quantities of interest. Furthermore, in the latter case, an estimate is required on how accurate the force coefficients are approximated, i.e., an *a posteriori* error estimate that quantifies the error on the numerical solution measured in terms of the quantity of interest.

In the following Section 4.1 we outline the approach of *a posteriori* error estimation and adjoint-based mesh refinement for single target quantities. Then, in Section 4.2 we generalize this approach to multiple target quantities. In Section 4.4 we derive residual-based indicators which are targeted at resolving all flow features. Finally, in Section 4.5 we give a collection of numerical examples.

4.1 Error estimation and mesh refinement for single target quantities

We begin by recalling the general approach of duality based *a posteriori* error estimation for *single* target functionals; see e.g. [23, 52, 58] among many others. Furthermore, we give the standard algorithm, as described in e.g. [21, 58], of goal-oriented (adjoint-based) adaptive mesh refinement tailored to the accurate and efficient computation of a single target quantity.

Let us consider the nonlinear problem

$$\mathbf{N}\mathbf{u} = 0 \quad \text{in } \Omega, \qquad\qquad \mathbf{B}\mathbf{u} = 0 \quad \text{on } \Gamma, \tag{67}$$

where $\Omega \in \mathbb{R}^d$, $d > 1$, is an open bounded domain with boundary $\Gamma = \partial\Omega$. $\mathbf{N}$ is a nonlinear differential operator and $\mathbf{B}$ is a possibly nonlinear boundary operator on Γ. Let $\mathcal{N} : \mathbf{V} \times \mathbf{V} \to \mathbb{R}$ be a semi-linear form, nonlinear in its first argument and linear in its second argument, such that the nonlinear problem (67) is discretized as follows: find $\mathbf{u}_h \in \mathbf{V}_{h,p}$ such that

$$\mathcal{N}(\mathbf{u}_h, \mathbf{v}_h) = 0 \quad \forall \mathbf{v}_h \in \mathbf{V}_{h,p}. \tag{68}$$

Furthermore, let us assume that the discretization (68) is *consistent*, i.e., the analytical solution $\mathbf{u} \in \mathbf{V}$ satisfies the following equation:

$$\mathcal{N}(\mathbf{u}, \mathbf{v}) = 0 \quad \forall \mathbf{v} \in \mathbf{V}. \tag{69}$$

Here, $\mathbf{V}$ is some suitably chosen function space including the analytical solution $\mathbf{u} \in \mathbf{V}$ to the primal problem (67) and satisfying $\mathbf{V}_{h,p} \subset \mathbf{V}$, where $\mathbf{V}_{h,p}$ is a discrete function space on the mesh $\mathcal{T}_h = \{\kappa\}$ consisting of elements κ covering the computational domain Ω; cf. [7, 54]

for the choice of $\mathbf{V}$ in the case of DG methods. Subtracting (69) from (68) we then obtain the Galerkin orthogonality

$$\mathcal{N}(\mathbf{u}, \mathbf{v}_h) - \mathcal{N}(\mathbf{u}_h, \mathbf{v}_h) = 0 \quad \forall \mathbf{v}_h \in \mathbf{V}_{h,p}. \tag{70}$$

Let $J(\cdot)$ be a nonlinear and differentiable target functional. We define the mean–value linearization of $J(\cdot)$ as follows

$$\bar{J}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h) = J(\mathbf{u}) - J(\mathbf{u}_h) = \int_0^1 J'[\theta\mathbf{u} + (1-\theta)\mathbf{u}_h](\mathbf{u} - \mathbf{u}_h)\,\mathrm{d}\theta, \tag{71}$$

where $J'[\mathbf{w}](\cdot)$ denotes the Fréchet derivative of $J(\cdot)$ evaluated at some $\mathbf{w}$ in $\mathbf{V}$.

Analogously, for $\mathbf{v}$ in $\mathbf{V}$, we define the mean–value linearization of $\mathcal{N}(\cdot, \mathbf{v})$

$$\begin{aligned} \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{v}) &= \mathcal{N}(\mathbf{u}, \mathbf{v}) - \mathcal{N}(\mathbf{u}_h, \mathbf{v}) \\ &= \int_0^1 \mathcal{N}'[\theta\mathbf{u} + (1-\theta)\mathbf{u}_h](\mathbf{u} - \mathbf{u}_h, \mathbf{v})\,\mathrm{d}\theta. \end{aligned} \tag{72}$$

Here, $\mathcal{N}'[\mathbf{w}](\cdot, \mathbf{v})$ denotes the Fréchet derivative of $\mathbf{u} \mapsto \mathcal{N}(\mathbf{u}, \mathbf{v})$, for $\mathbf{v} \in \mathbf{V}$ fixed, at some $\mathbf{w}$ in $\mathbf{V}$. Let us now introduce the following adjoint problem: find $\mathbf{z} \in \mathbf{V}$ such that

$$\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{w}, \mathbf{z}) = \bar{J}(\mathbf{u}, \mathbf{u}_h; \mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V}. \tag{73}$$

Choosing $\mathbf{w} = \mathbf{u} - \mathbf{u}_h$ in (73), recalling the linearization performed in (71), and exploiting the Galerkin orthogonality (70) we get

$$\begin{aligned} J(\mathbf{u}) - J(\mathbf{u}_h) &= \bar{J}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h) = \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{z}) \\ &= \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) = -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) \quad \forall \mathbf{z}_h \in \mathbf{V}_{h,p}. \end{aligned}$$

Thereby, we have the following error representation formula

$$J(\mathbf{u}) - J(\mathbf{u}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h), \tag{74}$$

where $\mathcal{R}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) = -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h)$ includes the primal residuals multiplied by the difference of the adjoint solution $\mathbf{z}$ and an arbitrary discrete function $\mathbf{z}_h \in \mathbf{V}_{h,p}$, see the definition of $\mathcal{R}(\cdot, \cdot)$ for the compressible Euler and Navier-Stokes equations in (44) and (64), respectively.

We note that the error representation formula (74) depends on the unknown analytical solution $\mathbf{z}$ to the adjoint problem (73) which in turn depends on the unknown analytical solution $\mathbf{u}$ to the primal problem (67). Thus, in order to render these quantities computable, both $\mathbf{u}$ and $\mathbf{z}$ must be replaced by suitable approximations. Here, the linearizations leading to $\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \cdot, \cdot)$ and $\bar{J}(\mathbf{u}, \mathbf{u}_h; \cdot)$ are performed about $\mathbf{u}_h$ and the adjoint solution $\mathbf{z}$ is replaced by the solution $\bar{\mathbf{z}}$ to the following linearized adjoint problem: find $\bar{\mathbf{z}} \in \mathbf{V}$ such that

$$\mathcal{N}'[\mathbf{u}_h](\mathbf{w}, \bar{\mathbf{z}}) = J'[\mathbf{u}_h](\mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V}. \tag{75}$$

This is then approximated by the discrete adjoint problem: find $\bar{\mathbf{z}}_h \in \bar{\mathbf{V}}_{h,p}$ such that

$$\mathcal{N}'[\mathbf{u}_h](\mathbf{w}_h, \bar{\mathbf{z}}_h) = J'[\mathbf{u}_h](\mathbf{w}_h) \quad \forall \mathbf{w}_h \in \bar{\mathbf{V}}_{h,p}. \tag{76}$$

Here, $\bar{\mathbf{V}}_{h,p}$ is an *adjoint* finite element space from which the approximate adjoint solution $\bar{\mathbf{z}}_h$ is sought. We remark that $\bar{\mathbf{z}}_h$ should not be calculated using the same finite element space $\mathbf{V}_{h,p}$ employed for the primal problem; otherwise the resulting error representation formula would be identically zero. In practice, there are essentially three approaches to computing a numerical approximation $\bar{\mathbf{z}}_h$ of $\mathbf{z}$. The first approach is to keep the degree p of the approximating polynomial used to compute $\mathbf{u}_h$ fixed, but compute $\bar{\mathbf{z}}_h$ on a sequence of adjoint finite element meshes $\bar{\mathcal{T}}_{\bar{h}}$ which, in general, differ from the "primal meshes" $\mathcal{T}_h$. Alternatively, $\bar{\mathbf{z}}_h$ may be computed using piecewise discontinuous polynomials of degree $\bar{p}$, $\bar{p} > p$, on the same finite element mesh $\mathcal{T}_h$ employed for the primal problem. A variant of this second approach is to compute the approximate adjoint problem using the same mesh $\mathcal{T}_h$ and polynomial degree p employed for the primal problem and to patchwise extrapolate the resulting approximate adjoint solution $\bar{\mathbf{z}}_h \in \mathbf{V}_{h,p}$ to an adjoint solution $\bar{\mathbf{z}}_h \in \mathbf{V}_{2h,\bar{p}}$, $\bar{p} > p$. While this latter approach is the cheapest of the three methods, and is still capable of producing adaptively refined meshes specifically tailored to the selected target functional, the quality of the resulting approximate error representation formula may be poor. On the basis of extensive numerical experimentation, we prefer to compute $\bar{\mathbf{z}}_h \in \mathbf{V}_{h,\bar{p}}$, $\bar{p} = p + p_{\text{inc}}$, i.e., we set $\bar{\mathbf{V}}_{h,p} = \mathbf{V}_{h,\bar{p}}$.

Rewriting the error representation (74) as follows

$$\begin{aligned} J(\mathbf{u}) - J(\mathbf{u}_h) =& \mathcal{R}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) \\ =& \mathcal{R}(\mathbf{u}_h, \mathbf{z} - \bar{\mathbf{z}}) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}} - \bar{\mathbf{z}}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h), \end{aligned} \tag{77}$$

we see that replacing the adjoint solution $\mathbf{z}$ in (74) by the solution $\bar{\mathbf{z}}_h$ to the discrete adjoint problem (76), we obtain the following approximate error representation

$$J(\mathbf{u}) - J(\mathbf{u}_h) \approx \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h). \tag{78}$$

This corresponds to ignoring in (77) the error $\mathcal{R}(\mathbf{u}_h, \mathbf{z} - \bar{\mathbf{z}})$ due to the linearization of the adjoint problem and the error $\mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}} - \bar{\mathbf{z}}_h)$ due to the approximation of the linearized adjoint problem. In fact, it can be shown (see e.g. [23]) that the linearization and the approximation errors of the adjoint problem are of higher order (quadratic) in the discretization error, $\mathbf{e} = \mathbf{u} - \mathbf{u}_h$, and may thus be neglected. In fact, in the series of publications [58, 59, 62], for example, among many others, it has been demonstrated that the approximate error representation in (78) is close to the true error in the target functional.

Finally, we note that (78) can be localized

$$J(\mathbf{u}) - J(\mathbf{u}_h) \approx \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h) \equiv \sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa, \tag{79}$$

where $|\bar{\eta}_\kappa|$ are local error indicators including the primal local residuals weighted with the discrete adjoint solution, denoted as adjoint-based indicators or as dual-weighted-residual (DWR) indicators, [23]. These local indicators can be used to drive an adaptive refinement (and coarsening) algorithm specifically tailored to the accurate and efficient approximation of the target quantity $J(\mathbf{u})$. For example, suppose that the aim of the computation is to compute $J(\cdot)$ such that the error $|J(\mathbf{u}) - J(\mathbf{u}_h)|$ is less than some user–defined tolerance `TOL`, i.e., $|J(\mathbf{u}) - J(\mathbf{u}_h)| \leq \texttt{TOL}$, then in practice we may enforce the stopping criterion $|\sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa| \leq \texttt{TOL}$. If this condition is not satisfied on the current finite element mesh $\mathcal{T}_h$,

then the local indicators η_κ are employed as local error indicators to guide mesh refinement and coarsening. The cycle of the goal-oriented adaptive mesh refinement [58] may be outlined as follows.

Algorithm 4.1 (Single-target adaptive algorithm) *Adaptive algorithm for the accurate and efficient approximation of a single target quantity $J(\mathbf{u})$:*

1. *Construct an initial mesh $\mathcal{T}_h$.*
2. *Compute $\mathbf{u}_h \in \mathbf{V}_{h,p}$, see (68), on the current mesh $\mathcal{T}_h$.*
3. *Compute $\bar{\mathbf{z}}_h \in \bar{\mathbf{V}}_{h,p}$, see (76), on the same mesh employed for $\mathbf{u}_h$, with $\bar{p} > p$.*
4. *Evaluate the approximate error representation $\mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h) = \sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa$.*
5. *If $|\sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa| \leq$ `TOL`, where `TOL` is a given tolerance, then `STOP`.*
6. *Otherwise, refine and coarsen a fixed fraction of the total number of elements according to the size of $|\bar{\eta}_\kappa|$ and generate a new mesh $\mathcal{T}_h$;* `GOTO` *2.*

Again, in several publications, e.g. [23, 52, 53, 59, 64], the versatility of this adaptive algorithm has been demonstrated.

4.2 Error estimation for multiple target quantities

In the following we present an extension of this approach to the efficient and accurate computation of *multiple* target quantities. Given, say N target quantities we replace the computation of N adjoint solutions as required in standard approaches by the solution of *two* auxiliary problems, namely one discrete adjoint problem and one discrete error equation where the latter can also be considered as the adjoint to the adjoint problem. In particular, the solution to the discrete error equation provides the *a posteriori* error estimation of arbitrary many target quantities. Furthermore, the solution to the adjoint problem related to an appropriately defined combination of the original target functionals provides the adjoint-based refinement indicators required for goal-oriented refinement.

This approach has been developed and applied to the scalar inviscid Burgers equation considering point values in [60]. It has later been extended to the treatment of laminar compressible flows considering multiple aerodynamic force coefficients in [55].

4.2.1 The standard approach

Let us now consider the extension of the above analysis to the error estimation and goal-oriented mesh refinement for multiple target quantities. Given N target functionals $J_i(\mathbf{u}), i = 1, \ldots, N$, the standard approach for deriving an error representation formula analogous to (74) for each $J_i(\cdot)$ is to introduce the following N adjoint problems: find $\mathbf{z}_i \in \mathbf{V}$ such that

$$\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{w}, \mathbf{z}_i) = \bar{J}_i(\mathbf{u}, \mathbf{u}_h; \mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{V}, \tag{80}$$

for $i = 1, \ldots, N$. Analogous to (74) we obtain the following error representation formulae

$$J_i(\mathbf{u}) - J_i(\mathbf{u}_h) = \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{z}_i - \mathbf{z}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{z}_i - \mathbf{z}_h), \tag{81}$$

for each $J_i(\cdot)$, $i = 1, \ldots, N$. In practice, the adjoint solutions $\mathbf{z}_i$, $i = 1, \ldots, N$, are unknown analytically and must be approximated numerically. After linearization and approximation we have: find $\bar{\mathbf{z}}_{i,h} \in \bar{\mathbf{V}}_{h,p}$ such that

$$\mathcal{N}'[\mathbf{u}_h](\mathbf{w}_h, \bar{\mathbf{z}}_{i,h}) = J_i'[\mathbf{u}_h](\mathbf{w}_h) \quad \forall \mathbf{w}_h \in \bar{\mathbf{V}}_{h,p}; \tag{82}$$

this amounts to solving N systems of linear equations with the same matrix but N different right–hand side vectors. Based on the discrete adjoint solutions $\bar{\mathbf{z}}_{i,h}$, $i = 1, \ldots, N$, the following approximate error representation formulae and local error indicators can be evaluated

$$J_i(\mathbf{u}) - J_i(\mathbf{u}_h) \approx \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_{i,h} - \mathbf{z}_h) = \sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa^{(i)}, \tag{83}$$

for $i = 1, \ldots, N$.

4.2.2 A new approach

In view of the error representation formula (81) an alternative approach consists of considering the following error equation: find $\mathbf{e} \in \mathbf{V}$ such that

$$\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{e}, \mathbf{w}) = \mathcal{R}(\mathbf{u}_h, \mathbf{w}) \qquad \forall \mathbf{w} \in \mathbf{V}, \tag{84}$$

whose solution is simply the discretization error $\mathbf{e} = \mathbf{u} - \mathbf{u}_h$. We remark that in the context of duality, (84) may be thought of as the adjoint of the adjoint problem and (81) the adjoint/adjoint-adjoint equivalence relating (80) to (84). Again after linearization, we obtain the following discrete error equation: find $\bar{\mathbf{e}}_h \in \bar{\mathbf{V}}_{h,p}$ such that

$$\mathcal{N}'[\mathbf{u}_h](\bar{\mathbf{e}}_h, \mathbf{w}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{w}_h) \quad \forall \mathbf{w}_h \in \bar{\mathbf{V}}_{h,p}. \tag{85}$$

Thereby, in practice, instead of solving N discrete adjoint problems, cf. (82), for $\bar{\mathbf{z}}_{i,h} \in \bar{\mathbf{V}}_{h,p}$ with data $J_i(\cdot)$ and then evaluating $\mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_{i,h} - \mathbf{z}_h)$ to determine the size of the error in the target functional $J_i(\cdot)$, $i = 1, \ldots, N$, one can simply solve the discrete error equation (84) for the approximate error $\bar{\mathbf{e}}_h \in \bar{\mathbf{V}}_{h,p}$ and evaluate

$$J_i(\mathbf{u}) - J_i(\mathbf{u}_h) = \bar{J}(\mathbf{u}, \mathbf{u}_h; \mathbf{e}) \approx J_i'[\mathbf{u}_h](\mathbf{e}) \approx J_i'[\mathbf{u}_h](\bar{\mathbf{e}}_h), \tag{86}$$

as an approximation to $J_i(\mathbf{u}) - J_i(\mathbf{u}_h)$, for $i = 1, \ldots, N$. When $N > 1$ this approach is clearly much more computationally efficient than the direct method. However, a disadvantage of this second approach is that while solving the discrete error equation (85) for $\bar{\mathbf{e}}_h$ gives information concerning the size of the error in the computed target functionals $J_i(\cdot)$, $i = 1, \ldots, N$, it does not provide the necessary local information on each element in the computational mesh to guide adaptive mesh refinement when the desired level of accuracy has not been achieved on the current mesh. On the other hand, computing the solution $\mathbf{z}_{i,h}$, $i = 1, \ldots, N$, to the N discrete adjoint problems (82), the approximate error representation formulae in (83) provide not only information concerning the size of the error in the computed target functionals, but also local error indicators $|\bar{\eta}_\kappa^{(i)}|$ which can be employed for adaptive mesh design.

4.3 Adaptive refinement for multiple target quantities

In this section we propose a strategy based on solving only two auxiliary problems (the discrete error equation (85) and an adjoint problem subject to appropriate data which stems from a specific combined target functional, cf. (91) below) which provide all the necessary information needed to both estimate the size of the error in the computed target functionals, as well as provide local error indicators that can be used to drive an adaptive mesh refinement algorithm.

Given N different target functionals $J_i(\cdot)$, $i = 1, \ldots, N$, $N > 1$, we would like to compute each $J_i(\mathbf{u}_h)$ to within a given user–defined tolerance $\texttt{TOL}_\texttt{i}$, $i = 1, \ldots, N$, respectively. More precisely, we consider the following problem: find $J_i(\mathbf{u}_h) \in \mathbb{R}$, $i = 1, \ldots, N$, such that

$$|J_i(\mathbf{u}) - J_i(\mathbf{u}_h)| \leq \texttt{TOL}_\texttt{i}\,, \quad \text{for } i = 1, \ldots, N. \tag{87}$$

However, as we want to define a combined target quantity $J_c(\cdot)$ including all original target quantities $J_i(\cdot)$, $i = 1, \ldots, N$, we weaken the requirement (87), and simply insist that the sum of the *relative* errors in each of the target functionals $J_i(\cdot)$, $i = 1, \ldots, N$, is less than $\texttt{TOL}$. In practice, since $J_i(\mathbf{u})$, $i = 1, \ldots, N$, is unknown, we approximate the sum of the relative errors by

$$\sum_{i=1}^{N} |J_i(\mathbf{u}) - J_i(\mathbf{u}_h)|/|J_i(\mathbf{u}_h)|, \tag{88}$$

see [60], assuming that $J_i(\mathbf{u}_h) \neq 0$, for $i = 1, \ldots, N$. As an alternative choice we might insist that the (weighted) sum of *absolute* errors in each of the target functionals $J_i(\cdot)$, $i = 1, \ldots, N$, is less than $\texttt{TOL}$, i.e., considering

$$\sum_{i=1}^{N} \alpha_i |J_i(\mathbf{u}) - J_i(\mathbf{u}_h)|. \tag{89}$$

where $\alpha_i > 0$, $i = 1, \ldots, N$. Here, choosing $\alpha_i = 1$, $i = 1, \ldots, N$, represents the special case of considering the (unweighted) sum of absolute errors.

Let us begin by assuming that the sign of the error in each target functional $J_i(\cdot)$, $i = 1, \ldots, N$, is known. For example, in some applications it may be known from either theoretical considerations or numerical experimentation that under mesh refinement the computed quantity of interest $J_i(\mathbf{u}_h)$ is always either smaller or greater than the exact value $J_i(\mathbf{u})$, for $i = 1, \ldots, N$. This includes the special case of monotonically convergent target quantities. For the case that under mesh refinement the quantity $J(\mathbf{u}_h)$ converges to $J(\mathbf{u})$ from above, for example, then the error $J(\mathbf{u}) - J(\mathbf{u}_h)$ is always negative; analogously, when it converges from below the error is always positive.

Employing this *a priori* knowledge concerning the convergence of the target functionals, we introduce a combined target functional

$$J_c(\mathbf{v}) = \sum_{i=1}^{N} \omega_i J_i(\mathbf{v})\,, \tag{90}$$

where $\omega_i = s_i/|J_i(\mathbf{u}_h)|$ or $\omega_i = \alpha_i s_i$, depending on whether the relative and weighted absolute errors (88) and (89), respectively, are considered. Here, s_i denotes the expected signs of the

errors $J_i(\mathbf{u}) - J_i(\mathbf{u}_h)$, $i = 1, \ldots, N$, respectively. Thereby, we may now proceed as in Section 4.1 to derive an error representation formula for the error in the combined target functional $J_c(\cdot)$. To this end, we introduce the following adjoint problem: find $\mathbf{z}_c \in \mathbf{V}$ such that

$$\mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{w}, \mathbf{z}_c) = \bar{J}_c(\mathbf{u}, \mathbf{u}_h; \mathbf{w}) \qquad \forall \mathbf{w} \in \mathbf{V}, \tag{91}$$

where $\bar{J}_c(\mathbf{u}, \mathbf{u}_h; \mathbf{w}) = \sum_{i=1}^{N} \omega_i \bar{J}_i(\mathbf{u}, \mathbf{u}_h; \mathbf{w})$ is the mean value linearization to J_c analogous to (71). Thus, we now deduce the following error representation formula

$$\begin{aligned} J_c(\mathbf{u}) - J_c(\mathbf{u}_h) &= \sum_{i=1}^{N} \omega_i (J_i(\mathbf{u}) - J_i(\mathbf{u}_h)) = \sum_{i=1}^{N} \omega_i \bar{J}_i(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h) \\ &= \mathcal{M}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \mathbf{z}_c - \mathbf{z}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{z}_c - \mathbf{z}_h) \end{aligned} \tag{92}$$

for all $\mathbf{z}_h \in \mathbf{V}_{h,p}$.

In general, the signs s_i, $i = 1, \ldots, N$, will not be known *a priori*. Thereby, we must first solve the discrete error equation (85) for $\bar{\mathbf{e}}_h$ and evaluate $\bar{s}_i = \operatorname{sgn}(J_i'[\mathbf{u}_h](\bar{\mathbf{e}}_h))$, $i = 1, \ldots, N$. Then, the adjoint problem (91) may be solved computationally using the predicted values of s_i, $i = 1, \ldots, N$, in $J_c(\cdot)$: find $\bar{\mathbf{z}}_{c,h} \in \bar{\mathbf{V}}_{h,p}$ such that

$$\mathcal{N}'[\mathbf{u}_h](\mathbf{w}_h, \bar{\mathbf{z}}_{c,h}) = J_c'[\mathbf{u}_h](\mathbf{w}_h) \quad \forall \mathbf{w}_h \in \bar{\mathbf{V}}_{h,p}. \tag{93}$$

Then the approximate error representation formula can be evaluated as follows

$$J_c(\mathbf{u}) - J_c(\mathbf{u}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{z}_c - \mathbf{z}_h) \approx \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_{c,h} - \mathbf{z}_h) \equiv \sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa. \tag{94}$$

This now provides both global information concerning the size of the error in the combined target functional $J_c(\cdot)$, as well as local information necessary for adaptive mesh refinement. Thus, the cycle of the adaptive algorithm can be outlined as follows.

Algorithm 4.2 (Multi-target adaptive algorithm) *Adaptive algorithm for the accurate and efficient approximation of multiple target quantities $J_i(\mathbf{u})$, $i = 1, \ldots, N$:*

1. *Construct an initial mesh $\mathcal{T}_h$.*
2. *Compute $\mathbf{u}_h \in \mathbf{V}_{h,p}$, see (68), on the current mesh $\mathcal{T}_h$.*
3. *Compute $\bar{\mathbf{e}}_h \in \bar{\mathbf{V}}_{h,p}$, see (85), on the same mesh employed for $\mathbf{u}_h$, with $\bar{p} > p$.*
4. *Evaluate $J_i(\mathbf{u}) - J_i(\mathbf{u}_h) \approx J_i'[\mathbf{u}_h](\bar{\mathbf{e}}_h) =: \psi_i$, $i = 1, \ldots, N$.*
5. *If $|\psi_i| \leq \texttt{TOL}_\texttt{i}$ for all $i = 1, \ldots, N$, then* `STOP`.
6. *Build the target quantity J_c based on $\bar{s}_i = sgn(\psi_i)$, $i = 1, \ldots, N$.*
7. *Compute $\bar{\mathbf{z}}_{c,h} \in \bar{\mathbf{V}}_{h,p}$, see (93), on the same mesh employed for $\mathbf{u}_h$, with $\bar{p} > p$.*
8. *Evaluate the approximate error representation $\sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa$, see (94).*
9. *If $|\sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa| \leq$ `TOL`, where `TOL` is a given tolerance, then* `STOP`.

10. Otherwise, refine and coarsen a fixed fraction of the total number of elements according to the size of $|\bar{\eta}_\kappa|$ and generate a new mesh $\mathcal{T}_h$; `GOTO` *2.*

Here, the stopping criterion in line (5) of Algorithm 4.2 corresponds to enforcing (87); on the other hand, the stopping criterion (9) corresponds to enforcing either equation (88) or (89) to be less than `TOL`, depending on the choice of weights in the combined target functional. This approach leads to the solution of only two auxiliary problems, in comparison to the N required for the standard approach.

We note that this approach has previously been developed for and applied to the DG discretization of the inviscid 1d Burgers equation in [60] considering the sum of relative errors of point values of the solution. In the following sections we apply this approach to the interior penalty DG discretization of the compressible Navier-Stokes equations [63] considering sums of relative and absolute errors of aerodynamic force coefficients including pressure induced and viscous drag, lift and moment coefficients.

4.4 Derivation of residual-based indicators

Provided the adjoint solution related to an arbitrary target functional is sufficiently smooth the corresponding error representation can be bounded from above by an error estimate (Type II error bound) which includes the primal residuals but is independent of the adjoint solution. By localizing this error estimate so-called residual-based indicators can be derived. Mesh refinement based on these indicators leads to meshes which resolve *all* flow features irrespective of any specific target quantity. We recall the derivation of them from [58, 62, 55]. Furthermore, we note that the residual-based indicators have been extended to include symmetry boundary conditions in [64].

Let $\mathbf{u}$ and $\mathbf{u}_h$ denote the solutions to (67) and (68), respectively. We now recall the error representation formula in (74), namely,

$$J(\mathbf{u}) - J(\mathbf{u}_h) = -\mathcal{N}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h), \tag{95}$$

for any $\mathbf{z}_h \in \mathbf{V}_{h,p}$. In particular, we can choose $\mathbf{z}_h := \Pi_h \mathbf{z} \in \mathbf{V}_{h,p}$ in (95), i.e.,

$$J(\mathbf{u}) - J(\mathbf{u}_h) = \mathcal{R}(\mathbf{u}_h, \mathbf{z} - \Pi_h \mathbf{z}), \tag{96}$$

where $\Pi_h \mathbf{z}$ denotes an appropriate interpolation/projection of $\mathbf{z}$ into the discrete function space $\mathbf{V}_{h,p}$. Indeed, here we select Π_h so that the following approximation property holds: given $\kappa \in \mathcal{T}_h$, suppose that $\mathbf{z}|_\kappa$ in $[H^{s_\kappa}(\kappa)]^5$, $0 \leq s_\kappa \leq p+1$. Then, there exists a constant C dependent on s_κ, p, and the shape regularity of $\mathcal{T}_h$, but is independent of the local mesh size h_κ, such that for $0 \leq m \leq s_\kappa$,

$$\|\mathbf{z} - \Pi_h \mathbf{z}\|_{H^m(\kappa)} \leq C h_\kappa^{s_\kappa - m} \|\mathbf{z}\|_{H^{s_\kappa}(\kappa)}. \tag{97}$$

Then, by employing the trace theorem, we have

$$\begin{aligned} \|\mathbf{z} - \Pi_h \mathbf{z}\|_{L_2(\partial\kappa)} &\leq C h_\kappa^{s_\kappa - 1/2} \|\mathbf{z}\|_{H^{s_\kappa}(\kappa)}, \qquad 1 \leq s_\kappa \leq p+1, \\ \|\mathbf{z} - \Pi_h \mathbf{z}\|_{H^1(\partial\kappa)} &\leq C h_\kappa^{s_\kappa - 3/2} \|\mathbf{z}\|_{H^{s_\kappa}(\kappa)}, \qquad 2 \leq s_\kappa \leq p+1; \end{aligned} \tag{98}$$

see Section 5.5; cf. also [8], for example. Using (64) we rewrite (96) as follows

$$\begin{aligned} J(\mathbf{u}) - J(\mathbf{u}_h) &= \int_\Omega \mathbf{R}(\mathbf{u}_h) \cdot (\mathbf{z} - \Pi_h \mathbf{z}) \, \mathrm{d}\mathbf{x} \\ &\quad + \sum_{\kappa \in \mathcal{T}_h} \int_{\partial\kappa \setminus \Gamma} \mathbf{r}(\mathbf{u}_h) \cdot (\mathbf{z} - \Pi_h \mathbf{z})^+ + \underline{\rho}(\mathbf{u}_h) : \nabla (\mathbf{z} - \Pi_h \mathbf{z})^+ \, \mathrm{d}s \\ &\quad + \int_\Gamma \mathbf{r}_\Gamma(\mathbf{u}_h) \cdot (\mathbf{z} - \Pi_h \mathbf{z})^+ + \underline{\rho}_\Gamma(\mathbf{u}_h) : \nabla (\mathbf{z} - \Pi_h \mathbf{z})^+ \, \mathrm{d}s, \end{aligned} \tag{99}$$

where the primal element residuals $\mathbf{R}(\mathbf{u}_h)$, the interior face residuals $\mathbf{r}(\mathbf{u}_h)$ and $\underline{\rho}(\mathbf{u}_h)$, and the boundary residuals $\mathbf{r}_\Gamma(\mathbf{u}_h)$ and $\underline{\rho}_\Gamma(\mathbf{u}_h)$ are given for the compressible Euler and Navier-Stokes equations in (45) and (65), respectively.

Assuming $\mathbf{z}|_\kappa \in [H^{s_\kappa}(\kappa)]^5$, $2 \leq s_\kappa \leq p+1$, for each $\kappa \in \mathcal{T}_h$, and applying Cauchy-Schwarz inequality and the approximation estimates (97) and (98) in (99) we obtain

$$|J(\mathbf{u}) - J(\mathbf{u}_h)| \leq \left(\sum_{\kappa \in \mathcal{T}_h} \left(\eta_\kappa^{(\mathrm{res})} \right)^2 \right)^{1/2}, \tag{100}$$

where $\eta_\kappa^{(\mathrm{res})}$ is given by

$$\eta_\kappa^{(\mathrm{res})} = h_\kappa^{s_\kappa} \|\mathbf{R}(\mathbf{u}_h)\|_{L_2(\kappa)} + h_\kappa^{s_\kappa - 1/2} \|\mathbf{r}_{\partial\kappa}(\mathbf{u}_h)\|_{L_2(\partial\kappa)} + h_\kappa^{s_\kappa - 3/2} \|\underline{\rho}_{\partial\kappa}(\mathbf{u}_h)\|_{L_2(\partial\kappa)}. \tag{101}$$

Here, we use the short notation $\mathbf{r}_{\partial\kappa} = \mathbf{r}$ on $\partial\kappa \setminus \Gamma$ and $\mathbf{r}_{\partial\kappa} = \mathbf{r}_\Gamma$ on Γ, i.e.,

$$\|\mathbf{r}_{\partial\kappa}(\mathbf{u}_h)\|^2_{L_2(\partial\kappa)} = \|\mathbf{r}(\mathbf{u}_h)\|^2_{L_2(\partial\kappa\setminus\Gamma)} + \|\mathbf{r}_\Gamma(\mathbf{u}_h)\|^2_{L_2(\Gamma)},$$

and analogously for $\underline{\rho}_{\partial\kappa}$, i.e.,

$$\|\underline{\rho}_{\partial\kappa}(\mathbf{u}_h)\|^2_{L_2(\partial\kappa)} = \|\underline{\rho}(\mathbf{u}_h)\|^2_{L_2(\partial\kappa\setminus\Gamma)} + \|\underline{\rho}_\Gamma(\mathbf{u}_h)\|^2_{L_2(\Gamma)}.$$

We point out that the *a posteriori* error bound (100) places severe regularity constraints on the adjoint solution $\mathbf{z}$, which are typically not fulfilled in practice. On the basis of numerical experimentation, and stimulated by the estimate (100), we employ following so-called *residual-based indicators*

$$\eta_\kappa^{\mathrm{res}} = h_\kappa \|\mathbf{R}(\mathbf{u}_h)\|_{L_2(\kappa)} + h_\kappa^{1/2} \|\mathbf{r}_{\partial\kappa}(\mathbf{u}_h)\|_{L_2(\partial\kappa)} + h_\kappa^{-1/2} \|\underline{\rho}_{\partial\kappa}(\mathbf{u}_h)\|_{L_2(\partial\kappa)}, \tag{102}$$

in subsequent numerical examples.

4.5 Numerical examples

In this section we give several examples which shall illustrate and explain the structure of adjoint solutions. In particular, we explain the adjoint solution's role associated with information transport, error transport, as well as error accumulation in numerical simulations, which is a key ingredient of error estimation and goal-oriented adaptive mesh refinement. Indeed, we show the adjoint solutions for a variety of problems, and demonstrate the accuracy of the error estimation, as well as the performance of the adaptive mesh refinement algorithm.

In the first two examples, see Sections 4.5.1 and 4.5.2, we revisit standard test cases of inviscid flows, the Ringleb flow problem and the supersonic flow past a wedge, respectively. Then in a third example, see Section 4.5.3, we consider a supersonic flow around an unsymmetric airfoil. In order to track paths of information and error transport in these flows and to understand the structure of the adjoint solution and the resulting adaptive mesh refinement, for all three examples, we choose a particularly simple target quantity, namely the solution (one component of it only) at one specific point in the computational domain.

In Section 4.5.4 we consider the error estimation and adjoint-based mesh refinement for an aerodynamic force coefficient. In particular, we demonstrate the performance of Algorithm 4.1 applied to approximating the pressure induced drag coefficient C_{dp}, for a supersonic viscous flow around a NACA0012 airfoil. The accuracy of the error estimation is then compared in Section 4.5.5 to that for an inviscid flow. In particular, here the effect of the linearization and discretization of the adjoint problem and the dependence on the smoothness of the flow and adjoint solutions are discussed.

Finally, in Section 4.5.6 we present several numerical results demonstrating the performance of the error estimation and adjoint-based mesh refinement for the accurate and efficient approximation of *multiple* force coefficients as outlined in Sections 4.2 and 4.3.

We note that in each of the following computations we set $p = 1$ for the numerical approximation of the flow equations, i.e., $\mathbf{u}_h \in \mathbf{V}_{h,1}$, and $p = 2$ for the discretization of the adjoint problems, i.e., $\bar{\mathbf{z}}_h \in \mathbf{V}_{h,2}$, if not stated differently in the text.

4.5.1 Ringleb flow problem

As first example taken from [52, 58] we consider the solution of the two dimensional (2d) compressible Euler equations to the Ringleb flow problem, that is one of the few non-trivial problems of the 2d Euler equations for which a (smooth) analytical solution is known. For this case the analytical solution may be obtained using the hodograph transformation, see [29] or the appendix of [52]. This problem represents a transonic flow in a channel, see Figure 2, with inflow and outflow boundaries given by the lower and upper boundaries of the domain, and slip-wall boundaries, i.e., vanishing normal velocity $\mathbf{v} \cdot \mathbf{n} = 0$, on the left and right boundary. The solution to this flow problem is smooth but it is transonic with a small supersonic region near the lower right corner. We choose the target functional to be

$$J(u) = \rho(-0.4, 2).$$

We note that this target functional is singular in the sense that it leads to a considerably rough adjoint solution that mainly consists of a single spike transported in reverse direction of the flow, see Figure 3(a). The mesh produced using the adjoint-based indicators, see (79), is shown in Figure 3(b). Here, we see that the mesh is mostly concentrated in the neighborhood of the characteristic upstream of the point of interest. However, due to the elliptic nature of the flow in the subsonic region, a circular region containing the point of interest is also refined, together with a strip of elements in the vicinity of the wall on the right–hand side of the domain enclosing the supersonic region of the flow.

4.5.2 Supersonic flow past a wedge

In this example taken from [52, 58] we study the formation of an oblique shock when a supersonic flow is deflected by a sharp object or wedge (also called supersonic compression

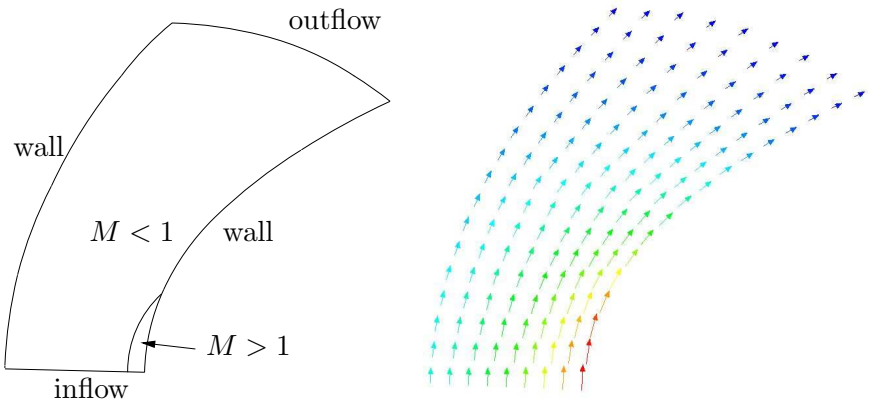
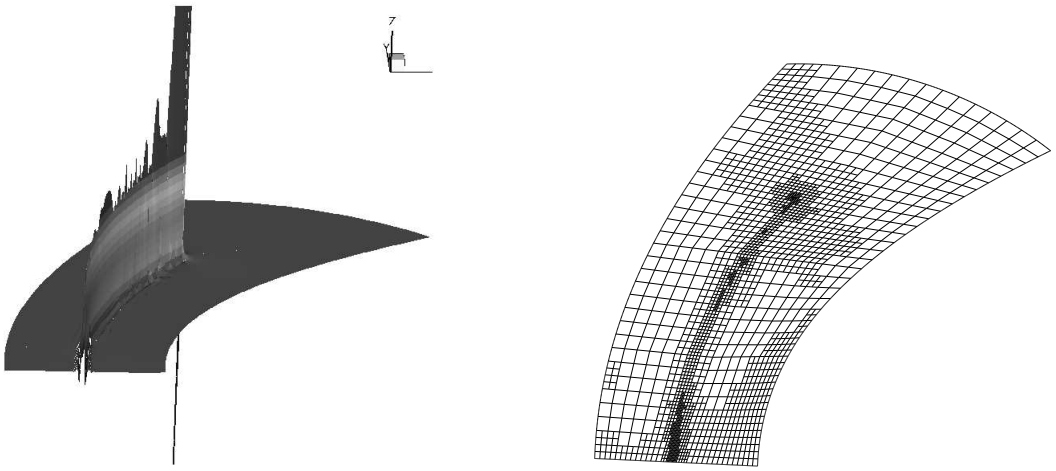


Figure 2: (a) Geometry for Ringleb's flow; M denotes the Mach number. (b) Flow direction coloured according to the Mach number.



(a) z_1 component of adjoint solution (b) adjoint-based refined mesh

Figure 3: Ringleb's flow problem, $J(u) = \rho(-0.4, 2)$, see [52]. Test case also considered in [58].

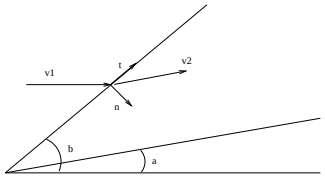


Figure 4: Geometry for the supersonic compression corner.

corner). Here, we consider a Mach 3 flow, over a compression corner of angle α which results in the development of a shock at an angle β, cf. Figure 4. By employing the Rankine–Hugoniot jump conditions, the analytical solution to this problem for a given α may be determined, see [5, 97] among others. Here, we select the wedge angle $\alpha = 9.5°$; thereby, the angle of the shock is given by $\beta = 26.9308°$. Furthermore, the true solution on the left– and right–hand side of the shock, in terms of conservative variables $(\rho, \rho v_1, \rho v_2, \rho E)$, are given by

$$\mathbf{u}_{\text{left}} \approx \begin{pmatrix} 1 \\ 3.5496 \\ 0 \\ 8.8 \end{pmatrix} \quad \text{and} \quad \mathbf{u}_{\text{right}} \approx \begin{pmatrix} 1.6180 \\ 5.2933 \\ 0.8858 \\ 13.8692 \end{pmatrix},$$

respectively.

Again, for simplicity, we consider a point evaluation; in particular, the point value

$$J(\mathbf{u}) = \rho(5, 2.05)$$

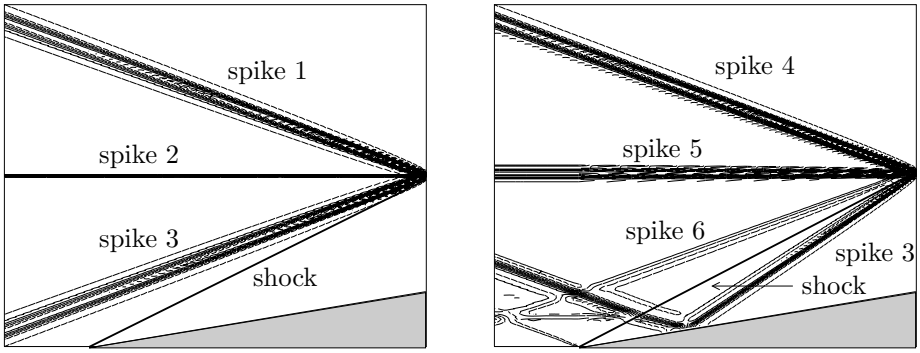
of the density just *in front of* the shock. In Figure 5(a) we show the z_1 component of the corresponding adjoint solution. It consists of three 'spikes', labelled 1, 2 and 3 in Figure 5(a), originating from the point of interest. These spikes are transported upstream along the characteristics corresponding to the three eigenvalues v and $v \pm c$, with $v = |\mathbf{v}| = \sqrt{v_1^2 + v_2^2}$ denoting the velocity of the gas and $c = \sqrt{\gamma p/\rho}$ the speed of sound. We note that the support of this adjoint solution does not intersect the region of the computational domain where the shock in the primal solution is located.

Let us now consider the more interesting case of a point evaluation of the density

$$J(\mathbf{u}) = \rho(5, 2.01)$$

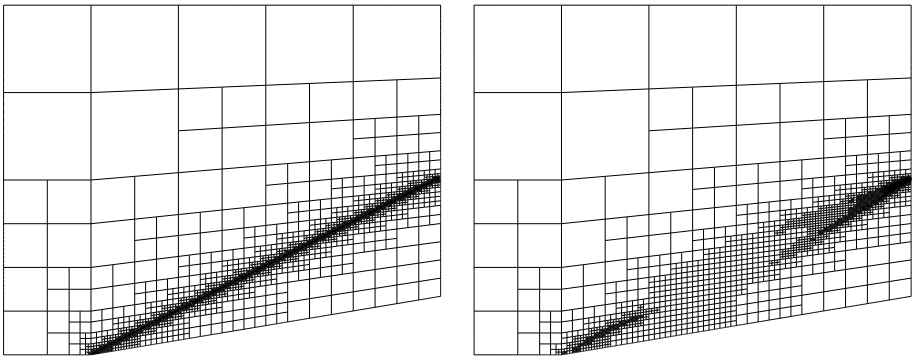
just *behind* the shock. Here, the support of the adjoint solution, see Figure 5(b), now intersects the region containing the shock and has a rather complicated structure. The two upper spikes of the adjoint solution both cross the shock in the neighborhood of the point of evaluation. At their crossing points they again each split into a further three spikes. These six spikes correspond to the three pairs of spikes, labelled spikes 4, 5 and 6 in Figure 5(b); the two spikes in each pair cannot be distinguished on the resolution shown, as they are extremely close together. Spike 3, corresponding to the same spike in Figure 5(a), is reflected off the inclined wall and crosses the shock at its bottom part.

A comparison of the adjoint solution in Figure 5(b) and the mesh in Figure 6(b), produced by the adjoint-based indicators (79) shows that the mesh has only been refined along the support of spikes 3 and 6 in the vicinity of the top part of the shock, and in the neighborhood



(a) (b)

Figure 5: Supersonic compression corner: z_1 component of adjoint solution for the supersonic compression corner for point evaluation of the density (a) in front of shock (b) behind shock.



(a) (b)

Figure 6: Supersonic compression corner, point evaluation of the density behind the shock: (a) residual-based refined mesh with 3821 elements ($|J(\mathbf{u}) - J(\mathbf{u}_h)| = 8.938 \times 10^{-3}$); (b) adjoint-based refined mesh constructed with 3395 elements ($|J(\mathbf{u}) - J(\mathbf{u}_h)| = 2.888 \times 10^{-4}$).

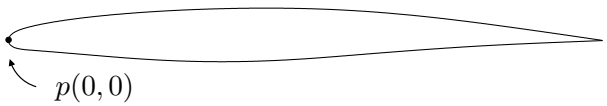


Figure 7: Profile of the BAC3-11 airfoil. Target quantity: pressure p at leading edge, [52, 58].

of the point where spike 3 crosses the bottom part of the shock. Comparing this mesh with the mesh in Figure 6(b) produced using residual-based indicators (102) we see that the adaptively refined meshes generated by employing the adjoint-based indicators are significantly more efficient than those produced using the residual-based indicators. Indeed, the true error in the computed target quantity is over an order of magnitude smaller when the adjoint-based indicators are employed.

This demonstrates that it is not necessary to refine the entire shock in this example to gain an accurate value of the target quantity under consideration, but only those parts that influence the target quantity either by material transport (eigenvalue v), or by information transported by the sound waves (eigenvalues $v \pm c$).

4.5.3 Supersonic flow past a BAC3-11 airfoil

In this example, taken from [52, 59], we study a supersonic flow around a BAC3-11 airfoil; this unsymmetric airfoil, see Figure 7, was originally specified in [3]. Here, we consider an inviscid flow at Mach number $M = 1.2$ and an angle of attack $\alpha = 5^\circ$.

The solution to this problem includes two shocks: one located in front of the leading edge of the airfoil and one originating from the trailing edge; see Figure 8(a) and also Figure 9(b) which shows a mesh that is refined at the position of the two shocks. Here, Figure 8(a) shows the Mach 1 isolines of the solution; the Mach $M = 1$ isoline to the left of the airfoil indicates the position of the first shock. The $M = 1$ isolines that originate from the upper and lower surfaces of the airfoil represent the transonic lines of the flow. The flow left of the first shock is supersonic; it is simply the $M = 1.2$ flow prescribed on the inflow boundary of the computational domain. The flow in between the shock and the transonic lines is subsonic; we note that the leading edge of the airfoil is located within this subsonic part of the flow. Finally, the flow behind the transonic lines is supersonic again.

In this example we take the target quantity to be the value of the pressure at the leading edge, i.e.,

$$J(u) = p(0,0),$$

cf. Figure 7. A computation on a fine mesh gives a reference value of $J(u) = 2.393$.

The structure of the solution $\bar{\mathbf{z}}_h$ to the discrete adjoint problem (76) corresponding to this point evaluation is displayed in Figure 8(b). This figure illustrates some principles of information transport in supersonic as well as in subsonic flow regions. To the right–hand side of the transonic lines the adjoint solution is zero as no information, neither by material transport nor even by information transport due to sound waves, can enter the subsonic region from the supersonic one. Within the whole subsonic region the adjoint solution is non-zero corresponding to the fact that sound waves can reach the point of evaluation from any point

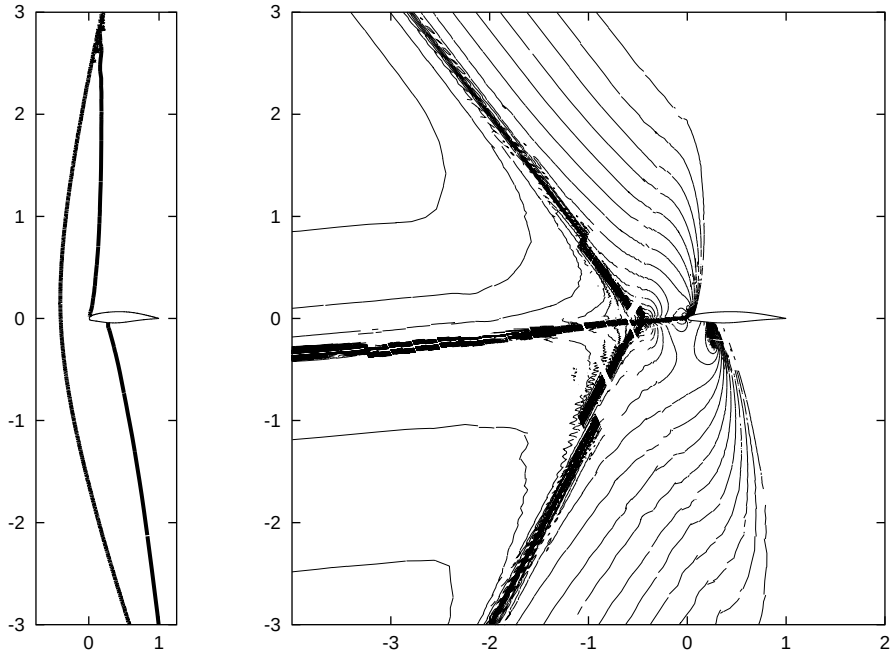
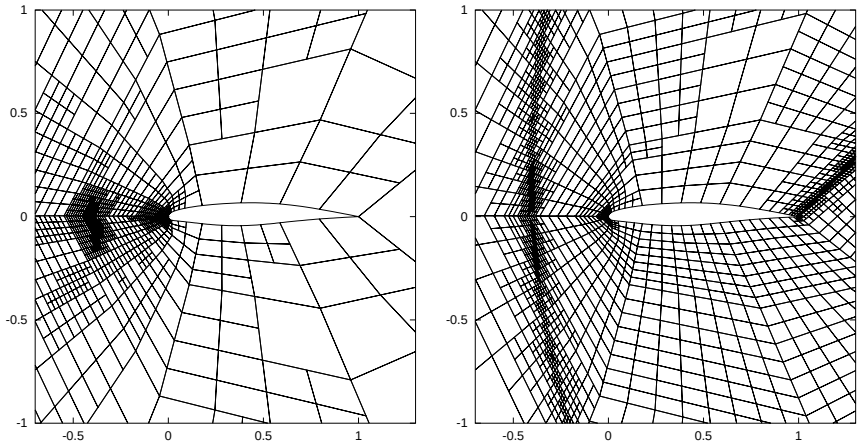


Figure 8: Supersonic BAC3-11 flow. (a) Mach 1 isolines of the flow solution; (b) z_1 isolines of adjoint solution, [52, 58].

in the subsonic area and that all numerical errors which occur within this subsonic region can (even though possibly to a small portion) affect the value of the solution at the point of evaluation. However, the adjoint solution in the subsonic region is concentrated in a thin spike that is transported upstream from the point of evaluation in direction of the flow. This spike corresponds to the path of material transport and represents the main path of information transport. To the left of the airfoil, this spike crosses the shock and splits into three spikes while entering the supersonic region left of the shock. These spikes are transported upstream along the characteristics corresponding to the three eigenvalues v and $v \pm c$. We recall that the characteristic corresponding to v represents the path of material transport, that in this example is given by the line inclined at 5 degrees, whereas the characteristics corresponding to $v \pm c$ represent the paths of information transport due to sound waves.

In Figure 9 we show the meshes produced using the adjoint-based and the residual-based error indicators. Here, we see that the mesh constructed using residual-based indicators is concentrated in the neighborhood of the two shocks. In contrast, the mesh produced using the adjoint-based indicators only refines the mesh in the vicinity of the point of evaluation and the part of the shock where the spike of the adjoint solution, i.e., where the main part of information, crosses the shock. The other parts of the shock are not resolved, as the numerical error in these regions only has a small affect on the accuracy of the solution at the point of evaluation. Also there is no refinement in the vicinity of the shock emanating from the trailing edge of the airfoil; thereby, this shock is not well resolved at all. Nevertheless, the solution at the leading edge of the airfoil is not affected by this as no information is transported upstream from the trailing edge, located in a supersonic part of the flow, to the



(a) (b)

Figure 9: Supersonic flow around BAC3-11 airfoil: (a) Adjoint-based refined mesh with 1803 elements ($|J(\mathbf{u}) - J(\mathbf{u}_h)| = 3.042 \times 10^{-3}$); (b) Residual-based refined mesh with 13719 elements ($|J(\mathbf{u}) - J(\mathbf{u}_h)| = 3.542 \times 10^{-2}$), [52, 58].

leading edge, located in the subsonic region. As in the previous example, we see that the adaptively refined meshes generated by employing the adjoint-based indicators are much more economical than those produced using the residual-based indicators. Indeed, in Figure 10 we clearly observe the superiority of the former error indicator; on the final mesh the true error in the computed functional is over two orders of magnitude smaller when the weighted error indicator is employed.

Motivated by the structure of the mesh generated by the adjoint-based error indicator, here we also consider the performance of an alternative *ad hoc* error indicator based on a modification of the residual indicator, whereby only elements in a neighborhood of a region upstream of the point of interest are marked for refinement. More precisely, we write C to denote the cone depicted in Figure 11(a) with apex half angle β, located in the center of the airfoil with symmetry axes inclined at $\alpha = 5^\circ$ according to the direction of the inflow. We now define the modified residual-based indicator $\eta_\kappa^{\mathrm{res,C}}$ as follows:

$$\eta_\kappa^{\mathrm{res,C}} = \begin{cases} \eta_\kappa^{\mathrm{res}}, & \text{if centroid}(\kappa) \in C, \\ 0, & \text{otherwise.} \end{cases}$$

This modification takes into account that we are not interested in the flow field in the whole domain, but only in the point value of the pressure at the leading edge. Thereby, adaptive mesh refinement is inhibited in the region downstream of the airfoil including the neighbourhood of the shock emanating from the trailing edge. Furthermore, refinement of the shock in front of the leading edge of the airfoil is prevented in regions that are placed too far above or below the airfoil since a low resolution of this shock in these areas is believed to not significantly degrade the accuracy of the pressure value at the leading edge, cf. Figure 9(a). In Figure 11(b) we show the mesh produced by employing $\eta_\kappa^{\mathrm{res,C}}$ with $\beta = 45^\circ$.

Finally, in Figure 10 we see that the modified residual indicator produces meshes that are much more efficient for computing the value of the pressure at the leading edge of the airfoil in comparison to the (unmodified) residual-based indicator $\eta_\kappa^{\mathrm{res}}$. Nevertheless, the meshes

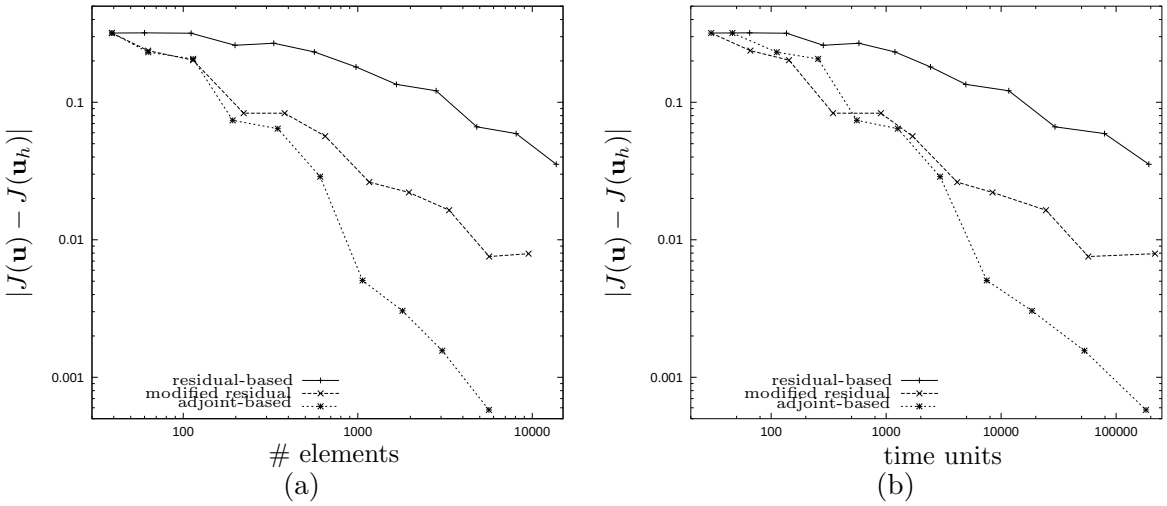


Figure 10: Supersonic flow around BAC3-11. Target quantity $J(\mathbf{u})$: pressure at leading edge. Use of the residual-based, the modified residual-based (*ad hoc*) and the adjoint-based indicators. Convergence of $|J(\mathbf{u}) - J(\mathbf{u}_h)|$ vs. (a) number of elements and (b) time units, [52].

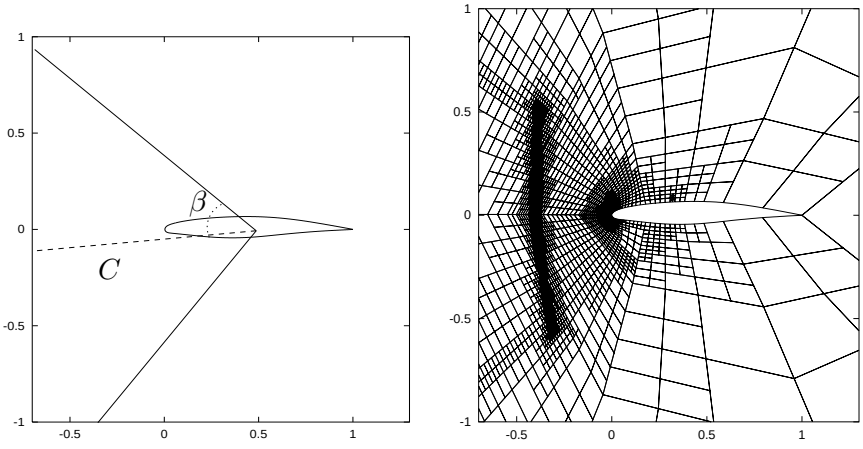
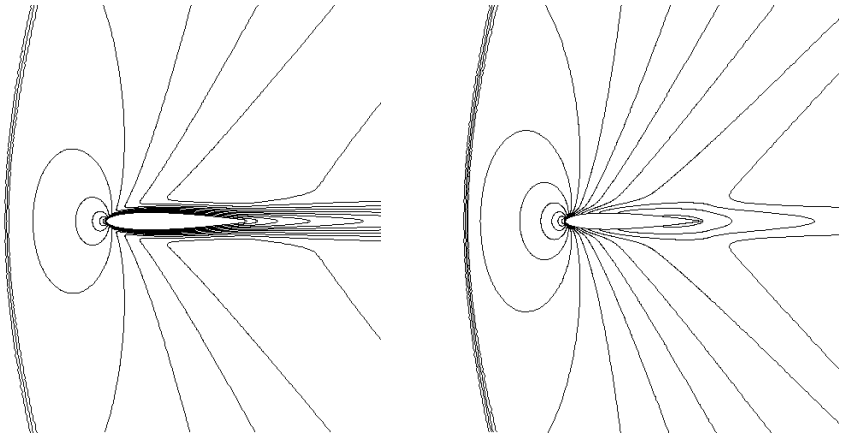


Figure 11: Supersonic flow around BAC3-11. (a) Cone C: domain where the modified residual (*ad hoc*) indicator is active; (b) Mesh constructed using the modified residual (*ad hoc*) indicator with 9516 elements ($|J(\mathbf{u}) - J(\mathbf{u}_h)| = 7.924 \times 10^{-3}$), [52, 58].



(a) (b)

Figure 12: Supersonic viscous flow: (a) Mach isolines and (b) density isolines, [53].

produced using the adjoint-based indicators are even more efficient than those designed by $\eta_\kappa^{\mathrm{res,C}}$; on the final mesh the true error in the computed functional is over an order of magnitude smaller when the adjoint-based error indicator is employed. We note that the chosen shape and size of the subdomain C and the resulting modified indicator only represents an 'attempt' to find a reasonable modification of the residual indicator $\eta_\kappa^{\mathrm{res}}$ that is capable of efficiently computing the pressure at the leading edge of the airfoil and to provide a 'fair' comparison with the goal–oriented adjoint-based indicator $|\bar{\eta}_\kappa|$. Indeed, the value of the angle β may be chosen differently, though *a priori* it is unclear which parts of the shock in front of the leading edge of the airfoil will influence the target functional. The angle β should not be chosen too small as otherwise the lack of resolution of the shock in front of the leading edge of the airfoil will impact on the computed value of the pressure at the point of interest; on the other hand choosing β too large may lead to over–refinement. In contrast, the adjoint-based indicator provides all the necessary information in order to decide which regions of the shock should be refined, and to what extent.

4.5.4 Supersonic viscous flow around the NACA0012 airfoil

In this example, taken from [53], we consider a horizontal viscous flow at $M = 1.2$ and $\mathrm{Re} = 1000$, with an adiabatic no-slip boundary condition imposed on the profile, see Figure 12. Due to the only slightly supersonic Mach number, the bow shock is located at some distance in front of the airfoil. Furthermore, there are two weak shocks emanating from the trailing edge of the airfoil, see Figure 13.

In the following we demonstrate that the approximate error representation $\mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h) = \sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa$, cf. (79), which was derived from the (exact) error representation (74) by replacing the (exact) adjoint solution $\mathbf{z}$ by a computed adjoint solution $\bar{\mathbf{z}}_h$, gives a good approximation to the true error measured in terms of the target quantity $J(\mathbf{u})$ under consideration. Furthermore, as in previous examples we highlight the advantages of designing an adaptive finite element algorithm based on adjoint-based indicators (79) in comparison to residual-based indicators (102).

Given that the flow is symmetric about the x-axis, both lift coefficients, C_{lp} and C_{lf},

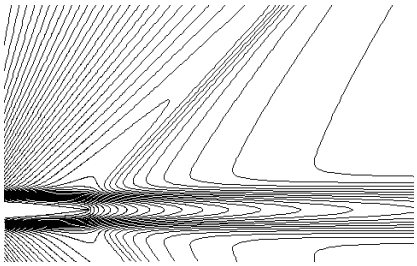


Figure 13: Supersonic viscous flow: Zoom of density isolines at trailing edge, [53].

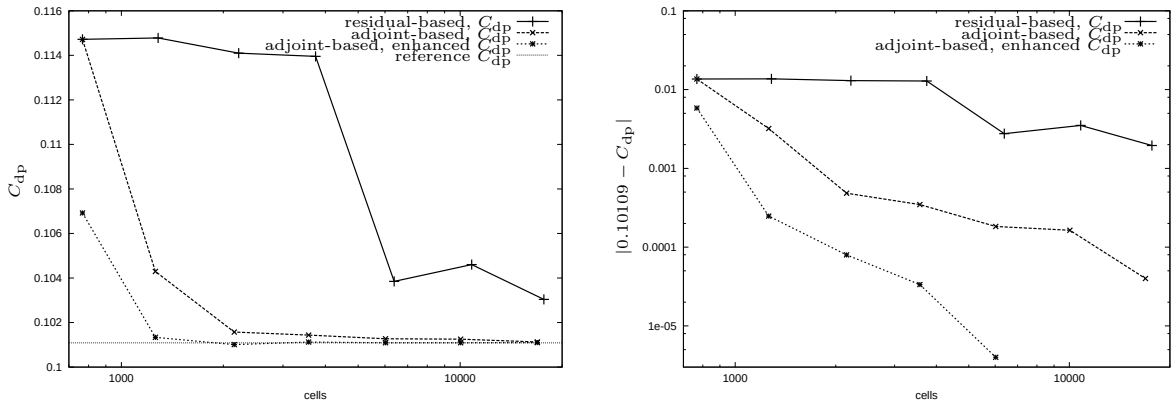
# Elements	# DoF	$J_{C_{\rm dp}}(\mathbf{u}) - J_{C_{\rm dp}}(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h}\bar{\eta}_\kappa$	θ
768	12288	-1.363e-02	-6.312e-03	0.46
1260	20160	-3.203e-03	-2.995e-03	0.94
2154	34464	-4.844e-04	-5.368e-04	1.11
3570	57120	-3.474e-04	-3.333e-04	0.96
6021	96336	-1.835e-04	-1.856e-04	1.01
10038	160608	-1.644e-04	-1.653e-04	1.01

Table 1: Supersonic viscous flow: Adaptive algorithm for the accurate approximation of $C_{\rm dp}$.

vanish. On the basis of fine grid computations the reference values of the pressure induced drag, $C_{\rm dp}$, and the viscous drag, $C_{\rm df}$, are given by $J_{C_{\rm dp}}(\mathbf{u}) \approx 0.10109$ and $J_{C_{\rm df}}(\mathbf{u}) \approx 0.10773$, respectively.

In the following, we consider the approximation of the pressure induced drag, $C_{\rm dp}$, i.e., the target quantity is $J(\cdot) = J_{C_{\rm dp}}(\cdot)$. In Table 1, we collect the data of the adaptive algorithm based on employing the adjoint-based indicators. Here, we show the number of elements and degrees of freedom (DoF) for $p = 1$ (bilinear elements), the true error in the target quantity, $J_{C_{\rm dp}}(\mathbf{u}) - J_{C_{\rm dp}}(\mathbf{u}_h)$, the approximate error representation formula $\mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h) := \sum_{\kappa\in\mathcal{T}_h}\bar{\eta}_\kappa$ and the effectivity index $\theta = \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)/(J_{C_{\rm dp}}(\mathbf{u}) - J_{C_{\rm dp}}(\mathbf{u}_h))$ of the error estimation. First, we note that on all meshes the right sign of the error is predicted, which is always negative in this computation, i.e., the computed $C_{\rm dp}$ values converge to the reference value from above. Furthermore, from the second mesh onwards, the approximate error representation represents a very good approximation to the true errors, which is indicated by the effectivity indices θ being very close to one.

In Figure 14 we compare the true error in the target quantity based on refining the computational mesh employing either the adjoint-based or residual-based indicators. Here, we see that for the first three refinement steps, when employing the residual-based indicator, the accuracy in the target quantity is hardly improved. In contrast to that, when using adjoint-based indicators, the error decreases significantly faster, being a factor of more than three smaller already after the second refinement step than the error on the finest residual-based refined mesh. Furthermore, the computed values of the target quantity $J(\mathbf{u}_h)$ can be enhanced by employing the approximate error representation $\mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h) = \sum_{\kappa\in\mathcal{T}_h}\bar{\eta}_\kappa$ to yield an enhanced value of the target quantity, $\tilde{J}_{C_{\rm dp}}(\mathbf{u}_h) = J_{C_{\rm dp}}(\mathbf{u}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)$. In Figure 14 we see, that the improved values, $\tilde{J}_{C_{\rm dp}}(\mathbf{u}_h)$, are significantly more accurate than the (baseline) $J_{C_{\rm dp}}(\mathbf{u}_h)$ values, and even show a higher rate of convergence. In fact, it can be shown, see



(a) (b)

Figure 14: Supersonic viscous flow: (a) $J_{C_{\rm dp}}(\mathbf{u}_h)$ values on residual-based refined meshes, $J_{C_{\rm dp}}(\mathbf{u}_h)$ and the enhanced values, $\tilde{J}_{C_{\rm dp}}(\mathbf{u}_h) = J_{C_{\rm dp}}(\mathbf{u}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)$, on adjoint-based refined meshes versus number of elements; (b) Error of these values versus number of elements, [53].

[62], that this value has a higher order of convergence than $J_{C_{\rm dp}}(\mathbf{u}_h)$, provided the primal and the adjoint solutions are smooth and the adjoint solution is approximated using higher-order polynomials. Furthermore, the approximate error representation is close to the true error even in cases of smooth adjoint solutions but possibly non-smooth primal solutions. The large difference in the performance, see Figure 14, of the adjoint-based indicator and the residual-based indicator in producing adaptively refined meshes for the accurate approximation of the target quantity $C_{\rm dp}$, is due to the very different parts of the computational meshes being marked for refinement by the two types of indicators. Figures 15 (a) & (b) show the finest mesh produced by employing the residual-based indicator. We see that this refinement criterion aims at resolving all flow features: the extensive bow shock, the wake of the flow behind the airfoil as well as the weak shocks emanating from the trailing edge of the airfoil. In contrast to that, the refinement of the mesh produced by employing the adjoint-based indicator, see Figures 15 (c) & (d), is very concentrated close to the airfoil. In particular, the bow shock is mainly resolved in a small region upstream of the profile only, and there is even no refinement at all at the position of the bow shock beyond six chord lengths above and below the profile. Furthermore, the weak shocks emanating from the trailing edge are not resolved and there is no refinement in the wake of the flow beyond three chord lengths behind the profile. Instead, the refinement of the mesh is concentrated near the leading edge of the profile and in the boundary layer of the flow. All other parts of the computational domain are recognized by the adjoint-based indicator to be of minor importance for the accuracy of the $C_{\rm dp}$ target quantity. In fact, the adjoint solution, see Figures 16 and 17, includes the crucial information concerning which local residuals contribute to the error in the target quantity and to what extent. Herewith, it offers all necessary information of error transport and accumulation. Finally, the adjoint-based indicators mark only those parts of the domain for refinement where residuals of the flow solution significantly contribute to the error of the target quantity, i.e., all parts which are important for the accurate approximation of the target quantity.

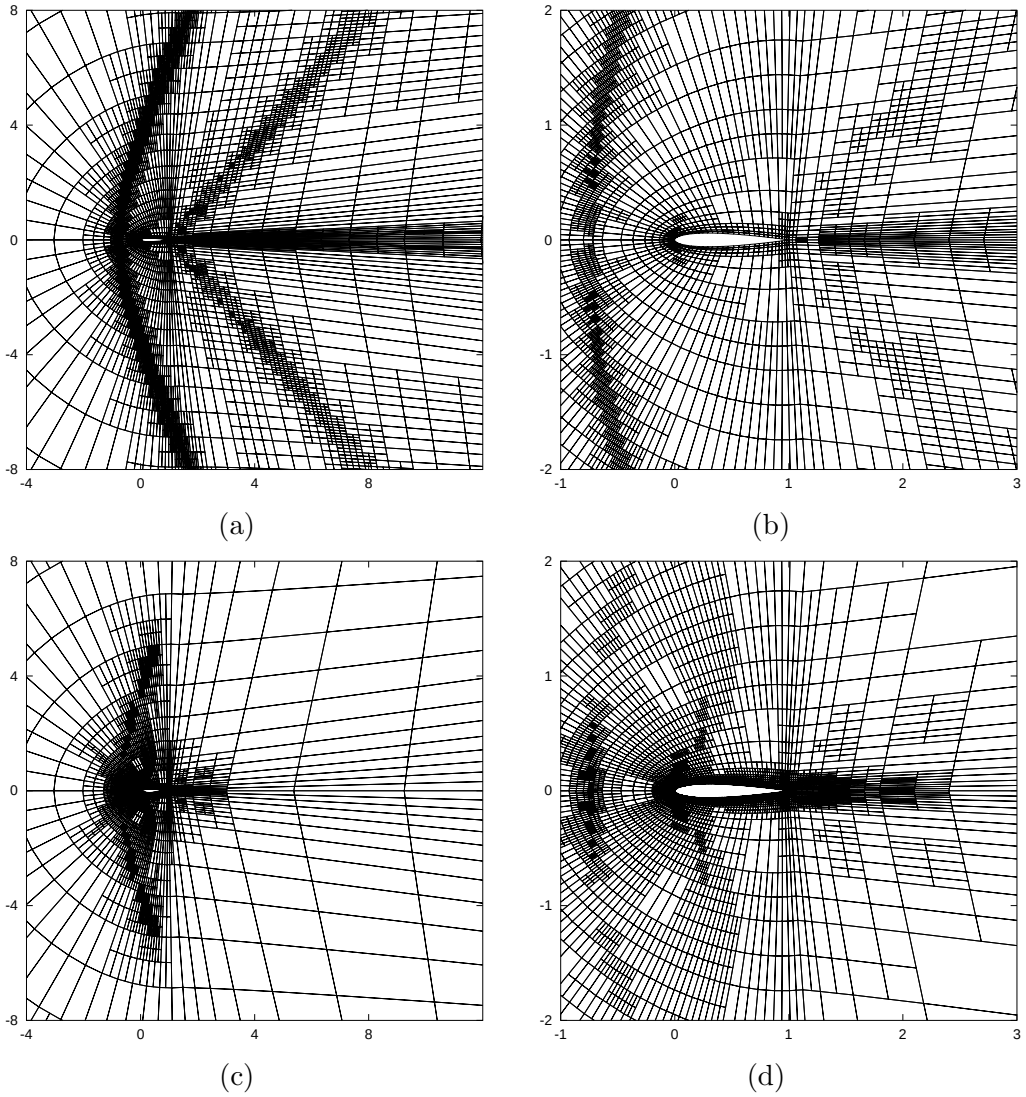
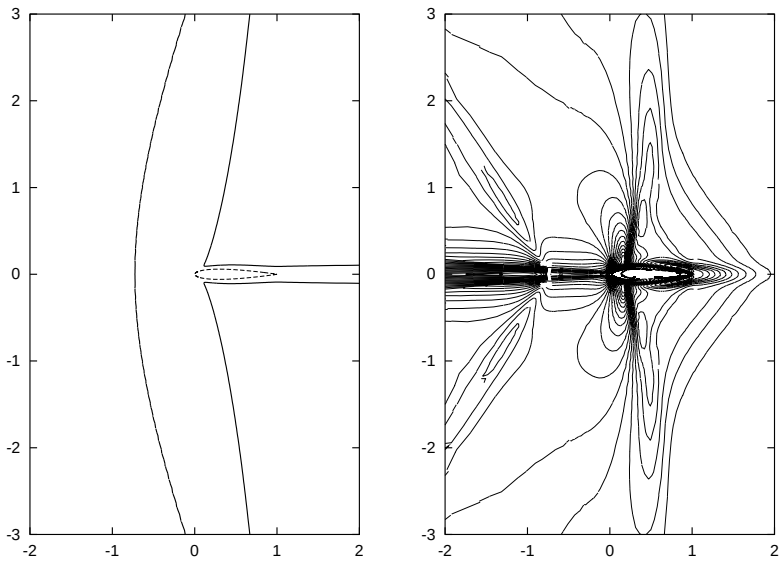


Figure 15: Supersonic viscous flow: (a) & (b) residual-based refined mesh of 17670 elements with 282720 degrees of freedom and $|J_{C_{\rm dp}}(\mathbf{u}) - J_{C_{\rm dp}}(\mathbf{u}_h)| = 1.9 \cdot 10^{-3}$; (c) & (d) goal-oriented refined mesh for $C_{\rm dp}$: mesh of 10038 elements with 160608 degrees of freedom and $|J_{C_{\rm dp}}(\mathbf{u}) - J_{C_{\rm dp}}(\mathbf{u}_h)| = 1.6 \cdot 10^{-4}$, [53].



(a) (b)

Figure 16: Supersonic viscous flow: (a) Sonic isolines of the flow solution; (b) isolines of the first component of the computed adjoint solution $\bar{\mathbf{z}}_h$, [53].

4.5.5 Comparison of the approximate error representation for viscous and inviscid flow.

We recall that the approximate error representation $\mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h) = \sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa$, cf. (79), was obtained by replacing the analytical solution $\mathbf{z}$ to the (exact) adjoint problem (73) in the error representation (74) by the solution $\bar{\mathbf{z}}_h$ to an approximate adjoint problem which is linearized about the discrete flow solution $\mathbf{u}_h$ and discretized. In order to discuss the error introduced by this replacement, we split the (exact) error representation (74) in three terms as follows:

$$\begin{aligned} J(\mathbf{u}) - J(\mathbf{u}_h) &= \mathcal{R}(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h) \\ &= \mathcal{R}(\mathbf{u}_h, \mathbf{z} - \bar{\mathbf{z}}) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}} - \bar{\mathbf{z}}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h), \end{aligned} \tag{103}$$

where the first term represents the error incurred through linearization of the adjoint problem, the second term is the error due to the numerical approximation of the (linearized) adjoint solution and the last term is the approximate error representation formula which is actually computed in practice. The error $\mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}} - \bar{\mathbf{z}}_h)$ due to the discretization of the adjoint problem will be of higher–order than the approximate error representation, provided that the adjoint solution is sufficiently regular and is approximated by higher order polynomials. The linearization error term $\mathcal{R}(\mathbf{u}_h, \mathbf{z} - \bar{\mathbf{z}})$ is expected to be small in cases when the analytical solution $\mathbf{u}$ is smooth. Rewriting the linearization term using $\mathcal{R}(\mathbf{u}_h, \mathbf{v}_h) = 0$ for any $\mathbf{v}_h \in \mathbf{V}_{h,p}$, we have that

$$\mathcal{R}(\mathbf{u}_h, \mathbf{z} - \bar{\mathbf{z}}) = \mathcal{R}(\mathbf{u}_h, (\mathbf{z} - \bar{\mathbf{z}}) - I_h(\mathbf{z} - \bar{\mathbf{z}})) = \mathcal{R}(\mathbf{u}_h, \mathbf{z} - I_h\mathbf{z}) - \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}} - I_h\bar{\mathbf{z}}), \tag{104}$$

where $I_h\mathbf{z} \in \mathbf{V}_{h,p}$ denotes a discrete approximation of $\mathbf{z}$. Here, we see that the linearization term can also be expected to be small when the adjoint solution is smooth.

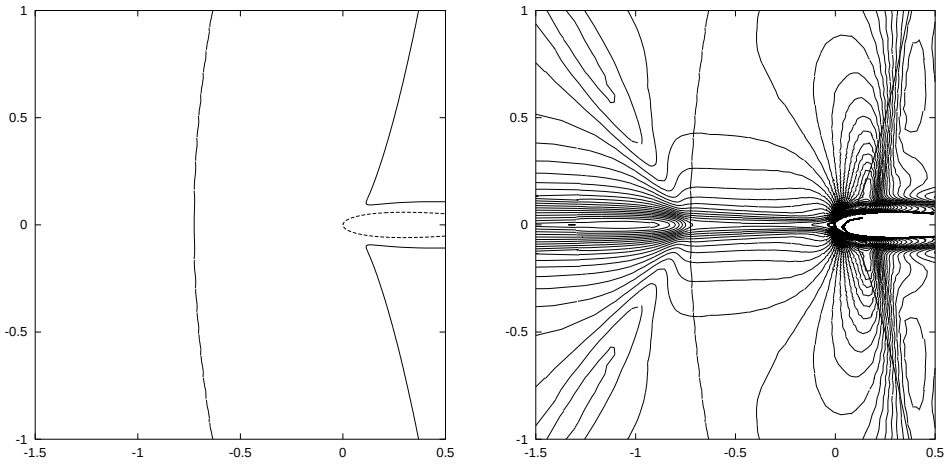


Figure 17: Supersonic viscous flow: Zoom of (a) sonic ($M = 1$) isolines of the flow solution; (b) together with isolines of first component of the discrete adjoint solution $\bar{\mathbf{z}}_h$, [53].

We note that the supersonic flow considered in the example in the previous section includes an extensive bow shock where the solution $\mathbf{u}$ is not smooth. In fact, all information of the flow crosses the shock from upstream before reaching the airfoil where the force coefficients are evaluated. Vice verse, all information of the adjoint problem, travelling in opposite direction along the flow characteristics, crosses the shock from downstream. According to the discussion above and given that $\mathbf{u}$ is not smooth, the linearization error term, $\mathcal{R}(\mathbf{u}_h, \mathbf{z}-\bar{\mathbf{z}})$, can only be expected to be small, when the adjoint solution is smooth. In this case, also the discretization error of the adjoint solution, $\mathcal{N}(\mathbf{u}_h, \bar{\mathbf{z}} - \bar{\mathbf{z}}_h)$, will be small, provided the adjoint solution is approximated with higher–order polynomials.

As can be seen in Figures 16 and 17, the adjoint solution is in fact smooth in most parts of the domain. In particular, at the position of the shock where the linearization error of flow solution is large, the adjoint solution is smooth. This, as already discussed above, is necessary for the linearization error term and the discretization error of the adjoint solution to be small, and finally for the approximate error representation to be close to the true error in the target quantity.

In fact, as shown in Table 1 for the viscous flow case considered, the approximate error representation represents a remarkably close estimate of the true error in the target quantity. In particular, the accuracy of the error estimation presented in Table 1 is significantly better than that presented in [59], for the supersonic inviscid flow around a BAC3-11 airfoil with a target quantity representing a (regularized) point evaluation, see also Section 4.5.3. This difference clearly is attributed to both, a smaller linearization error of the flow solution due to a smoother solution at a viscous shock, in contrast to at an inviscid shock, and to a smaller discretization error of an adjoint solution which is smoother for an adjoint problem being connected to a target quantity, $J(\mathbf{u}) = J_{C_{\mathrm{dp}}}(\mathbf{u})$, given by an integration of flow variables over a line (profile), than the solution to an adjoint problem which is connected to a (regularized) point evaluation.

In order to give a direct comparison with the viscous flow example at $M = 1.2$, $Re = 1000$

# Elements	# DoFs	$J_{C_{\rm dp}}(\mathbf{u}) - J_{C_{\rm dp}}(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h}\bar{\eta}_\kappa$	θ
768	12288	-1.184e-02	-2.218e-03	0.19
1260	20160	-4.214e-03	-6.197e-03	1.47
2151	34416	-9.285e-04	-5.458e-04	0.59
3687	58992	-2.472e-04	-3.666e-04	1.48
6165	98640	-9.057e-05	-9.796e-05	1.08
10605	169680	-6.057e-05	-6.150e-05	1.02

Table 2: Supersonic inviscid flow: Adaptive algorithm for the accurate approximation of $C_{\rm dp}$.

and $\alpha = 0°$, we consider the corresponding inviscid test case, with $M = 1.2$, $\alpha = 0°$ and the $C_{\rm dp}$ target quantity, in the following. Given the same freestream flow conditions and the same target quantity, this comparison shall give us a closer insight to a possibly increased linearization and discretization error of the adjoint solution for the inviscid flow in comparison to the viscous flow problem.

Given the $C_{\rm dp}$ reference value for the inviscid computation based on fine grid computations to be $J_{C_{\rm dp}}(\mathbf{u}) \approx 0.09549$, the data of the adaptive refinement targeted at the accurate approximation of this value is given in Table 2. Here, we see that the approximate error representation $\sum_{\kappa\in\mathcal{T}_h}\bar{\eta}_\kappa$ is still reasonably close the true error. But, there is a significant difference in the range of effectivity indices θ, which in the inviscid case is about 0.6-1.5 from the second mesh onwards, see Table 2, whereas in the viscous case this is about 0.94-1.11, cf. Table 1. This difference in the accuracy of the approximate error representation can be attributed to the increased linearization error at the (inviscid) shock and to a significantly less smooth adjoint solution in comparison to the viscous flow case. In fact, in Figures 18 and 19 we see that there are discontinuities of the adjoint solution near the trailing edge of the profile due to the supersonic nature of the flow in this part of the domain. Furthermore, there are discontinuities evolving close to the sonic lines of the flow above and below the profile. In addition, we see a number of wiggles upstream of the airfoil which is are not observed in the adjoint solution to the viscous flow problem, see Figures 16 and 17. This additional roughness is introduced from the primal solution, which is smoothed-out by the numerical (and artificial) viscosity of the DG scheme only, and as being an inviscid flow solution, lacks of any physical smoothing introduced by the governing differential equations. This results in the respective adjoint solution being significantly more rough than the adjoint solution to the (smoother) viscous flow solution. Finally, the adjoint solution shows some wiggles right at the position of the shock. Here, we have a coincidence in place of a large linearization error of the flow solution and an oscillatory adjoint solution, which results in some of the approximate error representations in Table 2 being less close to the true error, which is also indicated by the respective effectivity indices θ noticeably differing from one.

4.5.6 Error estimation and adjoint-based refinement for multiple target quantities

In this section, taken from [55], we present several numerical results demonstrating the performance of the error estimation and adjoint-based mesh refinement for the accurate and efficient approximation of *multiple* force coefficients. To this end, we consider the MTC3 test case defined in the European project ADIGMA [82]: laminar compressible flow around a NACA0012 airfoil with inflow Mach number equal to 0.5, at an angle of attack $\alpha = 2°$, and

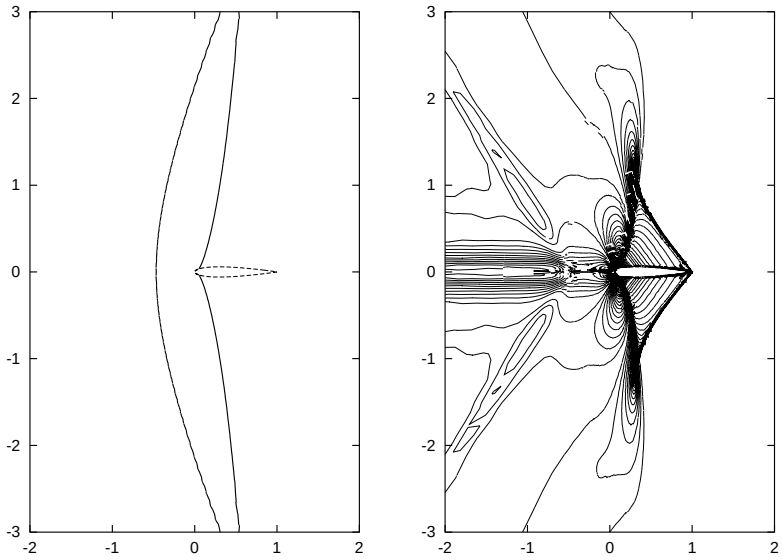


Figure 18: Supersonic inviscid flow: (a) Sonic isolines of the flow solution; (b) isolines of the first component of the discrete adjoint solution $\bar{\mathbf{z}}_h$, [53].

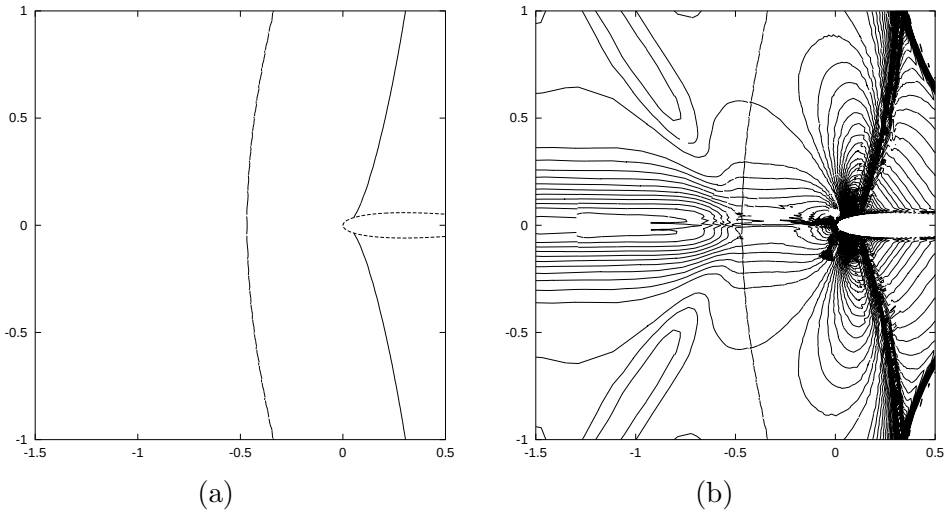


Figure 19: Supersonic inviscid flow: Zoom of (a) sonic ($M = 1$) isolines of the flow solution; (b) together with $\bar{z}_1$ isolines, [53].

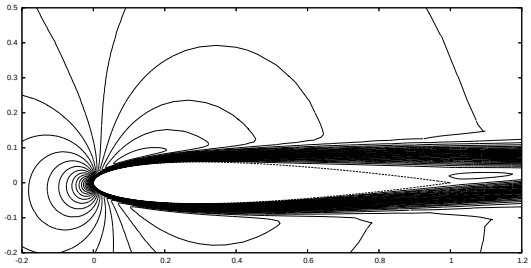


Figure 20: ADIGMA MTC3 test case: Mach number isolines. The laminar compressible flow at $M = 0.5$, $\alpha = 2°$, $\mathrm{Re} = 5000$ is a subsonic flow with a laminar separation at the trailing edge, [55].

Reynolds number $\mathrm{Re} = 5000$ with adiabatic no-slip wall boundary condition imposed on the airfoil geometry. This is a steady subsonic flow with a large laminar separation at the trailing edge, see Figure 20. The adaptive algorithms performed in the following will be based on the coarse mesh of 400 quadrilateral elements shown in Figure 21.

In this test case the most relevant aerodynamic force coefficients, namely the pressure induced and viscous drag coefficients, C_{dp} and C_{df}, respectively, the total lift coefficient C_{l} and the total moment coefficient C_{m} will be computed up to a predefined error tolerance $\mathtt{TOL}$. In the EU project ADIGMA the following industrial accuracy requirements have been defined for this test case:

$$\begin{aligned} |J_{C_{\mathrm{dp}}}(\mathbf{u}) - J_{C_{\mathrm{dp}}}(\mathbf{u}_h)| &\leq \mathtt{TOL}_{C_{\mathrm{dp}}} = 5 \cdot 10^{-4}, \\ |J_{C_{\mathrm{df}}}(\mathbf{u}) - J_{C_{\mathrm{df}}}(\mathbf{u}_h)| &\leq \mathtt{TOL}_{C_{\mathrm{df}}} = 5 \cdot 10^{-4}, \\ |J_{C_{\mathrm{l}}}(\mathbf{u}) - J_{C_{\mathrm{l}}}(\mathbf{u}_h)| &\leq \mathtt{TOL}_{C_{\mathrm{l}}} = 5 \cdot 10^{-3}, \\ |J_{C_{\mathrm{m}}}(\mathbf{u}) - J_{C_{\mathrm{m}}}(\mathbf{u}_h)| &\leq \mathtt{TOL}_{C_{\mathrm{m}}} = 5 \cdot 10^{-4}. \end{aligned} \tag{105}$$

Additionally, for academic purposes we define the following accuracy requirements:

$$|J_\star(\mathbf{u}) - J_\star(\mathbf{u}_h)| \leq \frac{1}{5}\mathtt{TOL}_\star, \qquad \text{for } \star \in \{C_{\mathrm{dp}}, C_{\mathrm{df}}, C_{\mathrm{l}}, C_{\mathrm{m}}\}, \tag{106}$$

where $\mathtt{TOL}_\star$ stands for the tolerances defined in (105). Thereby, the academic accuracy requirements are stronger in the sense that the tolerances for each of the force coefficients is a fifth of the tolerances for the industrial accuracy requirements.

We remark that in view of the discretization being adjoint consistent for specific force coefficients only, see Section 3.9 and [54], it would be preferable to approximate the total drag coefficient C_{d} rather than separately its pressure induced and viscous parts, C_{dp} and C_{df}, respectively. Nevertheless, in the case of wing design in industry, for example, some force coefficients are important to be accurately approximated separating pressure induced and viscous parts as is requested for the drag coefficient in this case.

Finally, we note that very fine grid computations, also with higher polynomial degrees, have been performed in order to obtain the following reference values (true values):

$$\begin{aligned} J_{C_{\mathrm{dp}}}(\mathbf{u}) &= C_{\mathrm{dp}}^{\mathrm{ref}} = 0.0238006, & J_{C_{\mathrm{df}}}(\mathbf{u}) &= C_{\mathrm{df}}^{\mathrm{ref}} = 0.0322805, \\ J_{C_{\mathrm{l}}}(\mathbf{u}) &= C_{\mathrm{l}}^{\mathrm{ref}} = 0.037468, & J_{C_{\mathrm{m}}}(\mathbf{u}) &= C_{\mathrm{m}}^{\mathrm{ref}} = -0.01662. \end{aligned} \tag{107}$$

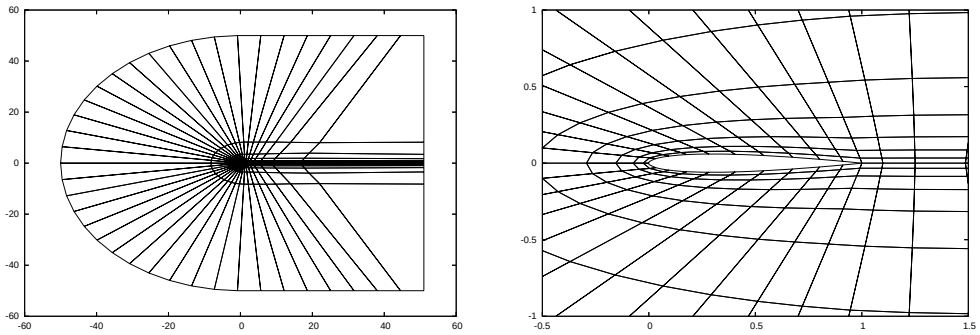


Figure 21: Coarse mesh with 400 elements: (left) full and (right) detailed view, [55].

These reference values will be used to compare with the force coefficients being evaluated on coarser meshes and using lower polynomial degrees in the following numerical examples. Also the accuracy of *a posteriori* error estimates will be investigated based on the reference values in (107).

In all subsequent computations we choose the penalization constant to be $C_{\text{IP}} = 20$ in (59). The solutions $\mathbf{u}_h$ to the nonlinear primal discretization (58) are computed in $\mathbf{V}_{h,p}$, with $p = 1$, i.e., the flow solutions are approximated using piecewise bilinear functions. By reducing the nonlinear residual over 6 orders of magnitude on each mesh, it is ensured that the resulting flow solutions are sufficiently converged such that iterative solver error contributions are negligible and errors observed with respect to force coefficients are due to the discretization only. As in [59, 62], for example, the solutions $\bar{\mathbf{z}}_h$ to the linear discrete adjoint problems (82) and (93) are computed in $\bar{\mathbf{V}}_{h,p} = \mathbf{V}_{h,\bar{p}}$ with $\bar{p} = p + 1$. Also the solutions $\bar{\mathbf{e}}_h$ to the discrete error equations (85) are computed in $\bar{\mathbf{V}}_{h,p} = \mathbf{V}_{h,\bar{p}}$.

In the following, we investigate the performance of the standard adaptive algorithm described in Section 4.1 and 4.2.1 in comparison to the proposed algorithm described in Sections 4.2.2 and 4.3.

The standard approach Given N target quantities, $J_i(\cdot)$, $i = 1, \ldots, N$, the standard approach of error estimation and goal-oriented adaptive mesh refinement consists of a multiple application of the single-target adaptive algorithm, i.e., the cycle of adaptive mesh refinement as given in Algorithm 4.1 is employed for each of the target quantities separately. This includes the solution of one discrete adjoint problem (82) for each of the target functionals, $J_i(\cdot)$, $i = 1, \ldots, N$, and the evaluation of the approximate error representation formulae (83) for $i = 1, \ldots, N$.

We note that this amounts to solving N systems of linear equations with the same matrix but N different right–hand side vectors. Although additional adjoint solutions may possibly be obtained cheaper by using an LU factorization of the matrix, for example, we refrain from this due to the memory requirements and use an iterative solver instead. Furthermore, a multiple application of Algorithm 4.1 leads to N separate sequences of adaptively refined meshes where each sequence is based on the same coarse grid but the subsequently refined meshes might differ from sequence to sequence. In fact, each of the N sequences of adaptively refined meshes is particularly tailored to the accurate approximation of one of the N target

# Elements	# DoF	$J_1(\mathbf{u}) - J_1(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa^{(1)}$	θ_1
400	6400	1.040e-03	-1.404e-03	-1.35
652	10432	3.347e-03	2.959e-03	0.88
1090	17440	4.105e-04	5.712e-04	1.39
1801	28816	-2.019e-04	-1.093e-04	0.54
3034	48544	-2.284e-04	-1.893e-04	0.83
5056	80896	-1.468e-04	-1.373e-04	0.94
8515	136240	-7.400e-05	-7.141e-05	0.96
14374	229984	-3.884e-05	-3.912e-05	1.01
24265	388240	-1.678e-05	-1.698e-05	1.01

Table 3: Single-target adaptive algorithm for the numerical approximation of $J_{C_{\rm dp}}(\mathbf{u})$.

# Elements	# DoF	$J_2(\mathbf{u}) - J_2(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa^{(2)}$	θ_2
400	6400	1.075e-02	1.525e-02	1.42
655	10480	-2.976e-03	-2.592e-03	0.87
1093	17488	-1.418e-03	-1.418e-03	1.00
1804	28864	-3.977e-04	-4.325e-04	1.09
2980	47680	-9.425e-05	-1.110e-04	1.18
5101	81616	-3.930e-05	-4.344e-05	1.11
8413	134608	-2.236e-05	-2.271e-05	1.02
14041	224656	-1.601e-05	-1.631e-05	1.02
23629	378064	-1.221e-05	-1.218e-05	1.00

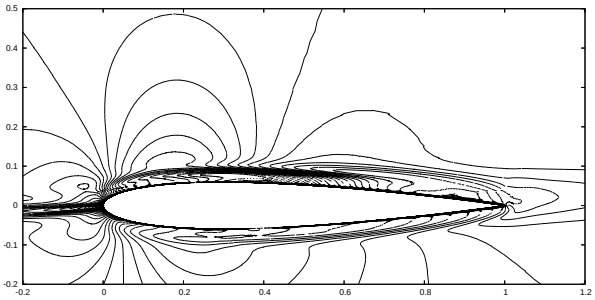
Table 4: Single-target adaptive algorithm for the numerical approximation of $J_{C_{\rm df}}(\mathbf{u})$.

quantities under consideration. As a consequence each of the adjoint problems must be solved on different meshes. In Tables 3, 4, 5 and 6 we demonstrate the performance of the standard approach for the numerical approximation of the pressure induced drag, the viscous drag, the total lift and the total moment coefficient, i.e.,

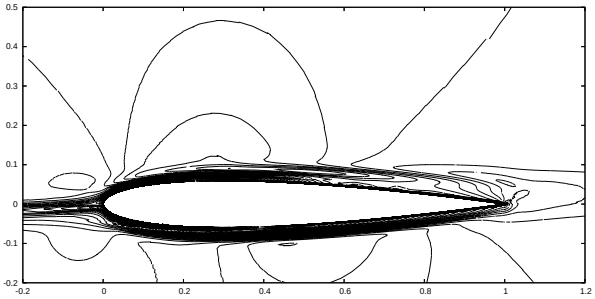
$$J_1(\mathbf{u}) = J_{C_{\rm dp}}(\mathbf{u}), \qquad J_2(\mathbf{u}) = J_{C_{\rm df}}(\mathbf{u}), \qquad J_3(\mathbf{u}) = J_{C_{\rm l}}(\mathbf{u}), \qquad J_4(\mathbf{u}) = J_{C_{\rm m}}(\mathbf{u}), \tag{108}$$

respectively. In each case, $i = 1, \ldots, 4$, we show the number of elements and degrees of freedom (DoF) in $\mathbf{V}_{h,1}$, the true error in the functional $J_i(\mathbf{u}) - J_i(\mathbf{u}_h)$, the approximate error representation formula $\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa^{(i)}$, and the corresponding effectivity index $\theta_i = \sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa^{(i)} / (J_i(\mathbf{u}) - J_i(\mathbf{u}_h))$. We see that on all meshes, excluding the initial coarse mesh, the quality of the computed error representation formulae $\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa^{(i)}$ is relatively good, in the sense that θ_i is close to one; moreover, as the mesh is refined, we observe that the effectivity indices θ_i improve by slowly tending towards unity.

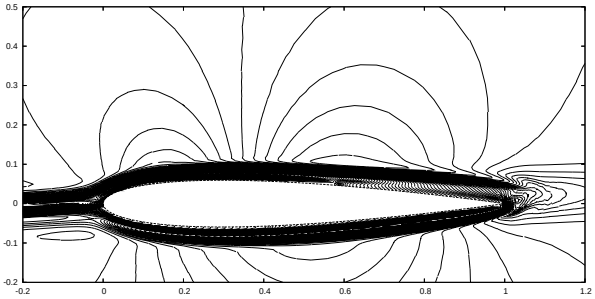
We note however, that for each target quantity one adjoint problem needs to be solved, see the z_1 components of the adjoint solutions related to the $C_{\rm dp}$, $C_{\rm df}$, $C_{\rm l}$ and $C_{\rm m}$ values in Figure 22. The four different adjoint solutions account for four different sensitivities of how local residuals contribute to the error in the respective target functionals under consideration. Based on this, four different sequences of meshes are created. The resulting locally refined meshes are particularly tailored to the accurate and efficient computation of the respective target quantity under consideration; for brevity we omit more details and refer to similar computations in, e.g. [62, 53]. The four sequences of meshes, however, amount to about four



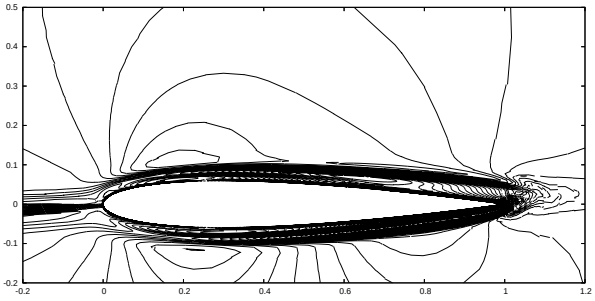
(a)



(b)



(c)



(d)

Figure 22: The z_1 components of the adjoint solution corresponding to the (a) C_{dp}, (b) C_{df}, (c) C_{l} and (d) C_{m} force coefficients.

# Elements	# DoF	$J_3(\mathbf{u}) - J_3(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa^{(3)}$	θ_3
400	6400	-1.173e-01	-5.869e-02	0.50
658	10528	6.730e-03	6.836e-03	1.02
1108	17728	-1.110e-03	-1.165e-03	1.05
1861	29776	-1.604e-03	-1.808e-03	1.13
3127	50032	-1.066e-03	-1.019e-03	0.96
5224	83584	-4.971e-04	-4.969e-04	1.00

Table 5: Single-target adaptive algorithm for the numerical approximation of $J_{C_\mathrm{l}}(\mathbf{u})$.

# Elements	# DoF	$J_4(\mathbf{u}) - J_4(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa^{(4)}$	θ_4
400	6400	-2.654e-03	-3.836e-03	1.45
670	10720	2.209e-03	2.055e-03	0.93
1138	18208	2.044e-04	1.647e-04	0.81
1912	30592	1.787e-05	1.910e-05	1.07
3295	52720	-1.704e-05	-1.693e-05	0.99

Table 6: Single-target adaptive algorithm for the numerical approximation of $J_{C_\mathrm{m}}(\mathbf{u})$.

times the number of flow solutions and adjoint solutions to be computed as compared to the case of considering one target functional only. This computational overhead increases as the number of target functionals N is increased rendering this approach inefficient for large N.

The new approach for multiple target functionals In this section we now employ the approach of adjoint-based error estimation and mesh refinement for multiple target quantities as proposed in Sections 4.2.2 and 4.3. Given N target quantities this approach of error estimation does not require N adjoint solutions. Instead, as described in Section 4.2.2, the solutions to N adjoint problems are replaced by the solution to *one* discrete error equation. Additionally, based on a combined target functional including all original target functionals, see Section 4.3, only one adjoint solution is required for obtaining adjoint-based indicators to be used in goal-oriented mesh refinement. In summary, this approach allows the error estimation and adjoint-based refinement based on two auxiliary problems, namely the discrete error equation and the discrete adjoint problem, irrespective of the number of target quantities.

Here, we consider the same test case as above; again, the goal is the accurate and efficient approximation of the pressure induced and the viscous drag, the total lift and the total moment coefficient, see (108), including providing error estimates for each of the computed quantities.

First we adopt the strategy of reducing the sum of the relative errors in the four target quantities, i.e., we choose the combined target functional in the adjoint problem (91) like in (90) with $\omega_i = s_i/|J_i(\mathbf{u}_h)|$, $i = 1, \ldots, 4$. This results in one sequence of adaptively refined meshes tailored to the accurate approximation of all four quantities. In Table 7 we collect the data of the adaptive algorithm. Here, we show the number of elements on the sequence of adaptively refined meshes. Furthermore, there are four columns, one for each of the target quantities $C_\mathrm{dp}, C_\mathrm{df}, C_\mathrm{l}$ and C_m. Each column is subdivided into two subcolumns where the left ones include the exact errors $J_i(\mathbf{u}) - J_i(\mathbf{u}_h)$, $i = 1, \ldots, 4$, and the right ones include the corresponding *a posteriori* error estimates $J_i'[\mathbf{u}_h](\bar{\mathbf{e}}_h)$, $i = 1, \ldots, 4$, see (86), based on the solution $\bar{\mathbf{e}}_h \in \bar{\mathbf{V}}_{h,p}$ to the discrete error equation (85). Here, we see that on all meshes,

	Error in C_{dp}		Error in C_{df}		Error in C_{l}		Error in C_{m}	
# Ele	Exact	Estim.	Exact	Estim.	Exact	Estim.	Exact	Estim.
400	1.0e-03	-2.8e-03	1.1e-02	1.7e-02	-1.2e-01	-6.6e-02	-2.7e-03	-4.3e-03
649	1.1e-03	1.2e-03	-3.0e-03	-2.9e-03	6.0e-03	3.7e-03	2.4e-03	2.0e-03
1114	-2.7e-04	-6.7e-05	-1.4e-03	-1.9e-03	-1.1e-03	-1.1e-03	3.8e-04	3.3e-04
1879	-4.2e-04	-3.3e-04	-6.2e-04	-7.5e-04	-6.6e-04	-1.0e-03	-4.5e-05	-9.0e-05
3163	-2.0e-04	-1.7e-04	-4.6e-04	-5.2e-04	-5.4e-04	-6.4e-04	-3.0e-05	-2.7e-05
5248	-1.4e-04	-1.2e-04	-2.3e-04	-2.6e-04	-3.9e-04	-5.7e-04	-8.8e-05	-9.3e-05

Table 7: Multi-target adaptive algorithm for the numerical approximation of C_{dp}, C_{df}, C_{l} and C_{m} targeted at the reduction of the sum of relative errors. The error estimation is based on the discrete error equation (85) and the estimate (86).

except the coarsest one, the estimated errors are quite close to the exact errors. In particular, the signs $\bar{s}_i = \mathrm{sgn}(J_i'[\mathbf{u}_h](\bar{\mathbf{e}}_h))$, $i = 1, \ldots, 4$, of the error estimates coincide with the signs $s_i = \mathrm{sgn}(J_i(\mathbf{u}) - J_i(\mathbf{u}_h)$, $i = 1, \ldots, 4$, of the respective exact errors. We recall that these signs are required in the definition of the combined target functional in (90) and are approximated by $\bar{s}_i$ as described in Section 4.3. We note that, here the difference between exact errors and error estimates are larger than the respective differences in the Tables 3-6. This is due to the fact that in Table 7 the estimates are based on (86) which includes two approximations: first the linearization of $J_i(\cdot)$ about the discrete function $\mathbf{u}_h$ and second the replacement of the exact error $\mathbf{e}$ by the solution $\bar{\mathbf{e}}_h \in \bar{\mathbf{V}}_{h,p}$ to the discrete error equation. In contrast, the error estimates in the Tables 3-6 include only one approximation, namely the replacement of the analytical adjoint solution $\mathbf{z}$ by the discrete adjoint solution $\mathbf{z}_h \in \bar{\mathbf{V}}_{h,p}$. Nevertheless, the estimates in Table 7 are sufficiently close to the exact errors to serve as reasonable indication of the size of the error in each target quantity. We recall that this error estimation is based not on four (or in general N) adjoint solutions like in Section 4.5.6 but based on one solution to a discrete error equation only.

Having investigated the accuracy of the *a posteriori* error estimation of the target quantities, we now concentrate on the accuracy of the evaluated target quantities achieved based on the two adaptive algorithms in this and the last subsection. Scanning through the Tables 3-6 we see that the industrial accuracy requirements (105) for C_{dp} (respectively, C_{df}, C_{l} and C_{m}) are reached after 2 (respectively, 3, 2 and 2) refinement steps. In contrast to that, in Table 7 we see that the accuracy requirements are reached after 2 (respectively, 4, 2 and 2) refinement steps. We notice that there is a slight increase in the number of refinement steps for the adaptive approach based on the combined target functional in Table 7 in comparison to the single-target adaptive approaches applied to each of the four of the target functionals separately, see Tables 3-6. As in single-target and multi-target optimization algorithms, this is due to the fact, that the single-target adapted meshes are optimized for the respective single target quantities whereas the multi-target adapted mesh is optimized for the combined target functional which results in a compromise between the single-target adapted meshes that cannot be as accurate for the individual target functionals as the respective single-target adapted meshes.

However, the efficiency of the adaptive mesh refinement can be improved: recalling that the accuracy requirements in (105) are given in terms of absolute errors where the tolerances of C_{dp}, C_{df} and C_{m} are 1/10 times the tolerance of C_{l}, we see that choosing the combined

# Ele	Error in C_{dp} Exact	Estim.	Error in C_{df} Exact	Estim.	Error in C_{l} Exact	Estim.	Error in C_{m} Exact	Estim.
400	1.0e-03	-2.8e-03	1.1e-02	1.7e-02	-1.2e-01	-6.6e-02	-2.7e-03	-4.3e-03
652	1.4e-03	1.4e-03	-3.0e-03	-2.9e-03	6.4e-03	4.1e-03	2.4e-03	2.0e-03
1138	-2.4e-04	-5.0e-05	-1.5e-03	-1.9e-03	-1.1e-03	-1.1e-03	4.3e-04	3.8e-04
1894	-4.7e-04	-3.2e-04	-2.9e-04	-4.9e-04	-5.1e-04	-8.4e-04	-5.5e-05	-6.2e-05
3112	-4.9e-05	2.6e-05	-4.0e-04	-5.0e-04	-5.6e-05	-2.6e-04	5.5e-05	6.3e-05
5131	-1.9e-04	-1.6e-04	-8.3e-05	-1.1e-04	-8.2e-04	-9.2e-04	-2.1e-05	-1.4e-05
8539	-1.0e-04	-8.1e-05	-2.2e-05	-4.9e-05	-1.1e-04	-3.3e-04	-2.4e-05	-1.8e-05

Table 8: Multi-target adaptive algorithm for the numerical approximation of C_{dp}, C_{df}, C_{l} and C_{m} targeted at the reduction of the weighted sum of absolute errors. The error estimation is based on the solution to the discrete error equation (85) and the estimate (86).

target functional $J_c(\cdot)$ to correspond to the weighted sum of absolute errors might be more appropriate for the problem at hand than the combined target functional corresponding to the sum of relative errors as used in Table 7. In fact, considering the weighted sum of absolute errors, i.e., $J_c(\cdot)$ is given by (90) with $\omega_i = \alpha_i s_i$, $i = 1, \ldots, 4$, and adjusting the weighting factors

$$\alpha_1 = 1, \qquad \alpha_2 = 1, \qquad \alpha_3 = 0.1, \qquad \alpha_4 = 1, \tag{109}$$

the influence of each target functional on the combined target functional corresponds to the specific accuracy requirements given in (105).

Analogous to the adaptive algorithm in Table 7 targeted at reducing the sum of relative errors, we now collect the corresponding data in Table 8 for the adaptive algorithm targeted at reducing the weighted sum of absolute errors. We see that the behaviour of the error estimation is similar to that described for Table 7. We recall that the latter two tables include the error estimates for the original force coefficients based on the solutions to the discrete error equations, see Figure 23. Additionally, for the combined target functional $J_c(\mathbf{u}_h)$ representing the weighted sum of absolute errors, we now collect the error estimates in Table 9. Here, we show the number of elements and degrees of freedom (DoF) in $\mathbf{V}_{h,1}$, the true error in the combined functional $J_c(\mathbf{u}) - J_c(\mathbf{u}_h)$, the approximate error representation formula $\sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa$, see (94), and the corresponding effectivity index $\theta_c = \sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa / (J_c(\mathbf{u}) - J_c(\mathbf{u}_h))$. We see that on all meshes, even including the initial coarse mesh, the quality of the computed error representation formulae $\sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa$ is extremely good in the sense that θ_c is close to one. We recall that these error estimates are based on the discrete adjoint solution (93) related to the combined target functional. Corresponding to the weighted sum (90) of the original target functionals, the adjoint solution $\mathbf{z}_c$, see Figure 24, represents a linear combination of the adjoint solutions, $\mathbf{z}_i$, $i = 1, \ldots, 4$, (depicted in Figure 22) which are related to the original target functionals $J_i(\mathbf{u})$, $i = 1, \ldots, 4$.

Considering again the accuracy of the computed target quantities we see in Table 8 that the industrial accuracy requirements (105) for C_{dp} (respectively, C_{df}, C_{l} and C_{m}) are reached after 2 (respectively, 3, 2 and 2) refinement steps which is equal to the number of refinement steps required in the respective single-target adaptive algorithms in Tables 3-6. However, we recall that the adaptive algorithms in Tables 3-6 include the solutions to four (or in general

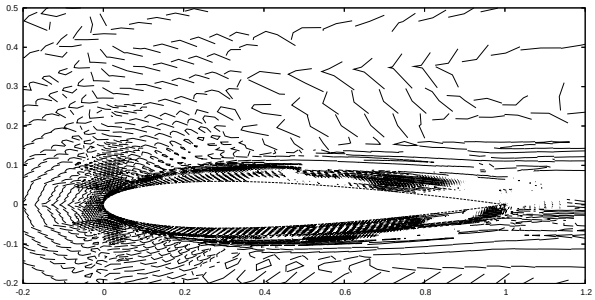


Figure 23: The first component of the solution $\bar{\mathbf{e}}_h$ to the discrete error equation, see (85), on the mesh of 8539 elements, cf. Table 8 and Figure 26(b).

# Elements	# DoF	$J_c(\mathbf{u}) - J_c(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa$	θ_c
400	6400	2.618e-02	2.636e-02	1.01
652	10432	7.378e-03	6.809e-03	0.92
1138	18208	2.258e-03	2.074e-03	0.92
1894	30304	8.582e-04	8.508e-04	0.99
3112	49792	5.087e-04	5.622e-04	1.11
5131	82096	3.753e-04	3.706e-04	0.99
8539	136624	1.622e-04	1.621e-04	1.00

Table 9: Multi-target adaptive algorithm for the approximation of C_{dp}, C_{df}, C_{l} and C_{m} targeted at the reduction of the weighted sum of absolute errors. The error estimation is based on the solution, see Figure 24 of the discrete adjoint problem (93) and the estimate (94).

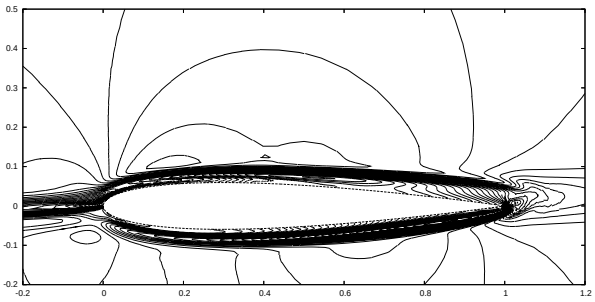
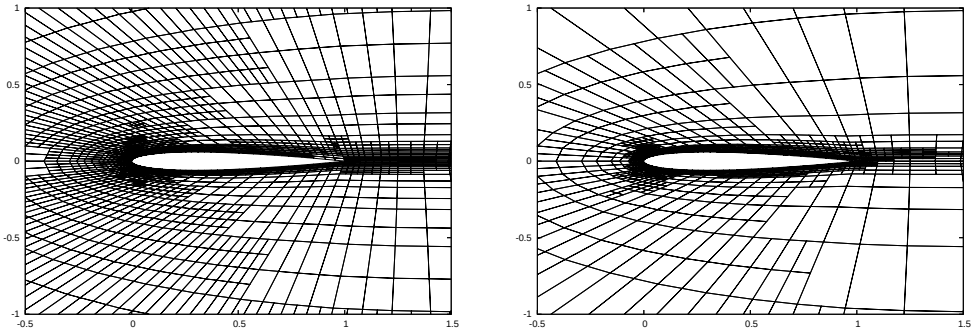


Figure 24: The z_1 components of the adjoint solution $\mathbf{z}_c$ corresponding to the combined target functional $J_c(\mathbf{u})$ which is related to the weighted sum of absolute errors.



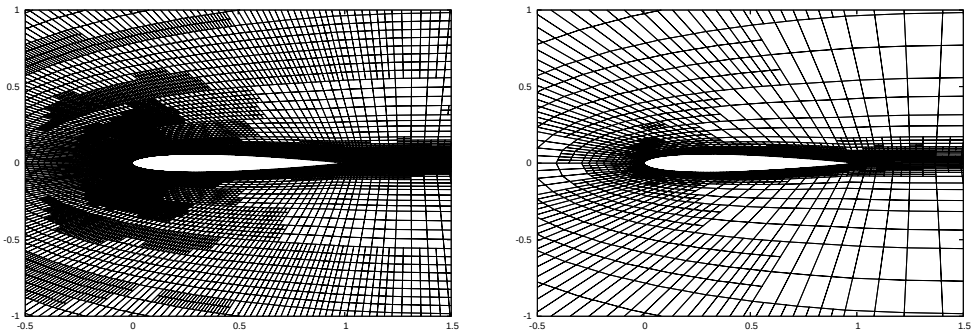
(a) (b)

Figure 25: The industrial accuracy requirements (105) are met on (a) the residual-based adapted mesh of 8896 elements in 149.4s and on (b) the multi-target adapted mesh of 1894 elements in 80.8s.

N) adjoint solutions, whereas the algorithm in Table 8 requires the solution to two auxiliary problems (the discrete error problem and the discrete adjoint problem) irrespective of the number of target functionals. This difference can also been seen in terms of computing times. In fact, the four separate single-target adaptive algorithms in Tables 3-6 add up to 147.5s to reach the industrial requirements whereas the multi-target adaptive algorithm, Table 8, requires 80.8s only. Note that this difference increases when considering N target quantities for $N > 4$. In fact, the computing time in the single-target adaptive algorithms can be expected to increase linearly with the number N of target quantities, whereas the computing time of the multi-target adaptive algorithm requires always two auxiliary problems to be solved and is thus independent of N.

Finally, we compare the adjoint-based (goal-oriented) adaptive algorithms discussed so far with an adaptive algorithm using the residual-based indicators 102. The indicators $|\eta_\kappa^{(\mathrm{res})}|$, $\kappa \in \mathcal{T}_h$, include primal residuals but no adjoint solution. In fact, these indicators are not targeted at the exact approximation of specific target quantities but at resolving all flow features. Given that they do not depend on the adjoint solution, the residual-based indicators are significantly faster to evaluate than the adjoint-based indicators. Nevertheless, as demonstrated in a sequence of earlier publications, [23, 52, 53, 59] among others, the sequences of meshes created using adjoint-based indicators are in general significantly more efficient and require much less computing resources for accurately approximating the target quantities under consideration than the meshes created using the residual-based indicators.

We observe now similar behaviour for the adjoint-based adaptive algorithm for multiple target functionals outlined in these lecture notes, in comparison to the residual-based algorithm. In fact, whereas the multi-target adaptive algorithm meets the industrial accuracy requirements (105) after 3 refinement steps with 1894 elements taking 80.8s, the residual-based adaptive algorithm meets the requirements after 6 refinement steps with 8896 elements in 149.4s; see the meshes in Figure 25. Note, however, that in the latter case no error estimates are available. In summary, we see that, even by including the computation of error estimates in each target quantity and in the combined target functional (weighted sum of relative errors) the multi-target adaptive algorithm is significantly more efficient than the residual-based algorithm.



(a) (b)

Figure 26: The academic accuracy requirements (106) are met on (a) the residual-based adapted mesh of 67660 elements in 2691.2s and on (b) the multi-target adapted mesh of 8539 elements in 664.63s, [55].

This difference becomes even more significant when the stronger accuracy requirements (106) are imposed. Scanning through Table 8 we see that these requirements for C_{dp} (respectively, C_{df}, C_{l} and C_{m}) are reached after 6 (respectively, 5, 3 and 3) refinement steps. In summary, using the multi-target algorithm the academic accuracy requirement are met after 6 refinement steps with 8539 elements in 664.63s, whereas using the residual-based algorithm the requirements are met after 10 refinement steps with 67660 elements in 2691.2s; see the meshes in Figure 26.

5 Development of anisotropic mesh adaptation

The mathematical modeling of advection, diffusion, and reaction processes arises in many application areas, not least aerodynamics. Typically, the diffusion is often small (compared to the magnitude of the advection and/or reaction), degenerate, or even vanishes in subregions of the domain of interest. This multi-scale behavior between the diffusion and the advection/reaction creates various challenges in the endeavor of computing numerical approximations to PDE problems of this type in an accurate and efficient manner. In particular, computationally demanding features may appear in the analytical solutions of such problems; these include boundary/interior layers or even discontinuities in the subregions where the problem is of hyperbolic type. When such, essentially lower-dimensional, features are present in the solution, the use of anisotropically refined meshes has been extensively advocated within the literature. Indeed, anisotropically refined meshes aim to be aligned with the domains of definition of these lower-dimensional features of the solution, in order to provide the necessary mesh resolution in the relevant directions, thereby reducing the number of degrees of freedom required to obtain an accurate approximation.

In this section, we consider the development of adaptive DG finite element methods based on employing anisotropically refined computational meshes. To this end, we shall first consider the application and analysis of DG methods for the numerical approximation of second–order partial differential equations with nonnegative characteristic form. This class of equations includes second–order elliptic and parabolic partial differential equations, ultra-parabolic equations, first–order hyperbolic problems, the Kolmogorov–Fokker–Planck equations of Brownian motion (see [12], for example), the equations of boundary layer theory in hydrodynamics, and various other degenerate elliptic equations. More generally, according to a well-known result of Hörmander [94], second–order hypoelliptic operators have nonnegative characteristic form at each point of the domain Ω, after possible multiplication by -1, so they too fall into this category.

The *a priori* error estimation presented here is based on exploiting the analysis developed in [50], which assumed that the underlying computational mesh is shape–regular, together with an extension of the techniques developed in [40] which precisely describe the anisotropy of the mesh; for related anisotropic approximation results, we refer to [6, 27, 40, 80, 83], for example. More specifically, we employ tools from tensor analysis, along with local singular-value decompositions of the Jacobi matrix of the local elemental mappings, to derive directionally-sensitive bounds for arbitrary polynomial degree approximations, thus generalizing the ideas presented in [40], where only the case of approximation with conforming linear elements was considered. The advantages of this general approach are that the resulting interpolation bounds exploit the full spectral properties of the underlying (affine) element transformation, and are thereby independent of the choice of coordinate axes. Additionally, no *a priori* condition on the maximal angle of the computational mesh is required; indeed, numerical experiments presented in [40] clearly demonstrate that this approach leads to approximation bounds which show the correct asymptotic behaviour with respect to the maximal angle. These interpolation error bounds are then employed to derive general anisotropic *a priori* error bounds for the DG approximation of linear functionals of the underlying analytical solution.

Additionally, *a posteriori* error bounds are derived based on the arguments outlined in Sections 2 & 4, cf. [22, 58, 77, 79], for example. On the basis of our *a posteriori* error

bound we design and implement an anisotropic adaptive algorithm to ensure the reliable and efficient control of the error in the prescribed target functional to within a given tolerance. This involves exploiting both local isotropic and anisotropic mesh refinement, based on choosing the most competitive subdivision of a given element κ from a series of trial (Cartesian) refinements. The superiority of the proposed algorithms in comparison with standard isotropic mesh refinement, and a Hessian–based anisotropic mesh refinement strategy, will be illustrated by a series of numerical experiments.

The discussion in this section represents a brief survey of the article [43]; see also [49].

5.1 Model problem and discretization

Let Ω be a bounded open polyhedral domain in $\mathbb{R}^d$, $d = 2, 3$, and let Γ signify the union of its $(d-1)$–dimensional open faces. We consider the advection–diffusion–reaction equation

$$Lu \equiv -\nabla \cdot (a\nabla u) + \nabla \cdot (\mathbf{b}u) + cu = f \ , \tag{110}$$

where $f \in L_2(\Omega)$ and $c \in L_\infty(\Omega)$ are real–valued, $\mathbf{b} = \{b_i\}_{i=1}^d$ is a vector function whose entries b_i are Lipschitz continuous real–valued functions on $\bar{\Omega}$, and $a = \{a_{ij}\}_{i,j=1}^d$ is a *symmetric* matrix whose entries a_{ij} are bounded, piecewise continuous real–valued functions defined on $\bar{\Omega}$, with

$$\boldsymbol{\zeta}^\top a(\mathbf{x})\boldsymbol{\zeta} \geq 0 \quad \forall \boldsymbol{\zeta} \in \mathbb{R}^d, \quad \text{a.e. } \mathbf{x} \in \bar{\Omega} \ . \tag{111}$$

Under this hypothesis, (110) is termed a *partial differential equation with nonnegative characteristic form*. By $\mathbf{n}(\mathbf{x}) = \{n_i(\mathbf{x})\}_{i=1}^d$ we denote the unit outward normal vector to Γ at $\mathbf{x} \in \Gamma$. On introducing the so called *Fichera function* $\mathbf{b} \cdot \mathbf{n}$ (cf. [94]), we define

$$\Gamma_0 = \left\{ \mathbf{x} \in \Gamma : \ \mathbf{n}(\mathbf{x})^\top a(\mathbf{x})\mathbf{n}(\mathbf{x}) > 0 \right\} ,$$
$$\Gamma_- = \{ \mathbf{x} \in \Gamma \backslash \Gamma_0 : \ \mathbf{b}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0 \} \ , \quad \Gamma_+ = \{ \mathbf{x} \in \Gamma \backslash \Gamma_0 : \ \mathbf{b}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) \geq 0 \} \ .$$

The sets Γ_- and Γ_+ will be referred to as the inflow and outflow boundary, respectively. Evidently, $\Gamma = \Gamma_0 \cup \Gamma_- \cup \Gamma_+$. If Γ_0 is nonempty, we shall further divide it into disjoint subsets $\Gamma_{\rm D}$ and $\Gamma_{\rm N}$ whose union is Γ_0, with $\Gamma_{\rm D}$ nonempty and relatively open in Γ. We supplement (110) with the boundary conditions

$$u = g_{\rm D} \quad \text{on} \quad \Gamma_{\rm D} \cup \Gamma_- \ , \qquad \mathbf{n} \cdot (a\nabla u) = g_{\rm N} \quad \text{on} \quad \Gamma_{\rm N} \ , \tag{112}$$

and adopt the (physically reasonable) hypothesis that $\mathbf{b} \cdot \mathbf{n} \geq 0$ on $\Gamma_{\rm N}$, whenever $\Gamma_{\rm N}$ is nonempty. Additionally, we assume that the following (standard) positivity hypothesis holds: there exists a constant vector $\boldsymbol{\xi} \in \mathbb{R}^d$ such that

$$c(\mathbf{x}) + \frac{1}{2} \nabla \cdot \mathbf{b}(\mathbf{x}) + \mathbf{b}(\mathbf{x}) \cdot \boldsymbol{\xi} > 0 \ \text{ a.e. } \mathbf{x} \in \Omega \ . \tag{113}$$

For simplicity of presentation, we assume throughout that (113) is satisfied with $\boldsymbol{\xi} \equiv \mathbf{0}$; we then define the positive function c_0 by

$$(c_0(\mathbf{x}))^2 = c(\mathbf{x}) + \frac{1}{2} \nabla \cdot \mathbf{b}(\mathbf{x}) \ \text{ a.e. } \mathbf{x} \in \Omega \ . \tag{114}$$

For the well-posedness theory (for weak solutions) of the boundary value problem (110), (112), in the case of homogeneous boundary conditions, we refer to [74, 78].

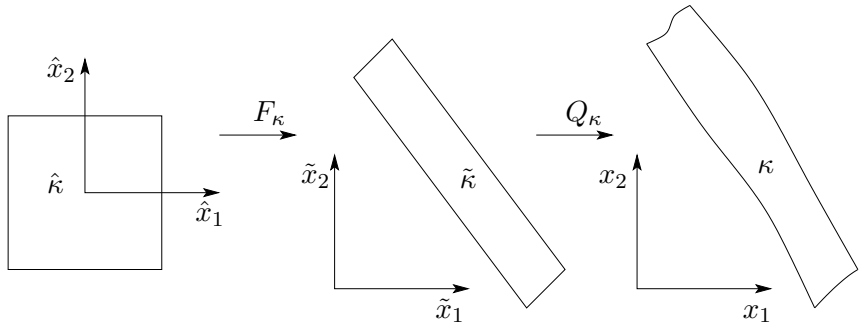


Figure 27: Construction of the element mapping via the composition of an affine mapping F_κ and a C^1–diffeomorphism Q_κ.

5.2 Meshes, finite element spaces and traces

Let $\mathcal{T}_h = \{\kappa\}$ be a subdivision of the (polyhedral) domain Ω into disjoint open element domains κ constructed through the use of the mappings $Q_\kappa \circ F_\kappa$, where $F_\kappa : \hat{\kappa} \to \tilde{\kappa}$ is an affine mapping from the reference element $\hat{\kappa}$ to $\tilde{\kappa}$, and $Q_\kappa : \tilde{\kappa} \to \kappa$ is a C^1–diffeomorphism from $\tilde{\kappa}$ to the physical element κ. Here, we shall assume that $\hat{\kappa}$ is either the hypercube $(-1,1)^d$ or the unit d–simplex; in the latter case Q_κ is typically the identity operator, unless curved elements are employed. The mapping F_κ defines the size and orientation of the element κ, while Q_κ defines the shape of κ, without any significant rescaling, or indeed change of orientation, cf. Figure 27 for the case when $d = 2$ and $\hat{\kappa} = (-1,1)^2$. With this in mind, we assume that the element mapping Q_κ is close to the identity in the following sense: the Jacobi matrix J_{Q_κ} of Q_κ satisfies

$$C_1^{-1} \leq \|\det J_{Q_\kappa}\|_{L_\infty(\kappa)} \leq C_1, \quad \|J_{Q_\kappa}^{-\top}\|_{L_\infty(\kappa)} \leq C_2, \quad \|J_{Q_\kappa}^{-\top}\|_{L_\infty(\partial\kappa)} \leq C_3 \tag{115}$$

for all κ in $\mathcal{T}_h$ uniformly throughout the mesh for some positive constants C_1, C_2, and C_3. This will be important as our error estimates will be expressed in terms of Sobolev norms over the element domains $\tilde{\kappa}$, in order to ensure that only the scaling and orientation introduced by the affine element maps F_κ are present in the analysis. Writing m_κ, $m_{\tilde{\kappa}}$, and $m_{\hat{\kappa}}$ to denote the d–dimensional measure of the elements κ, $\tilde{\kappa}$, and $\hat{\kappa}$, respectively, the above condition (115) implies that there exists a positive constant C_4 such that

$$C_4^{-1} m_{\tilde{\kappa}} \leq m_\kappa \leq C_4 m_{\tilde{\kappa}} \quad \forall \kappa \in \mathcal{T}_h. \tag{116}$$

The above maps are assumed to be constructed in such a manner to ensure that the union of the closure of the disjoint open elements $\kappa \in \mathcal{T}_h$ forms a covering of the closure of Ω, i.e., $\bar{\Omega} = \cup_{\kappa \in \mathcal{T}_h} \bar{\kappa}$. For a function v defined on κ, $\kappa \in \mathcal{T}_h$, we write $\tilde{v} = v \circ Q_\kappa$ and $\hat{v} = \tilde{v} \circ F_\kappa$ to denote the corresponding functions on the elements $\tilde{\kappa}$ and $\hat{\kappa}$, respectively. Thereby, we have that $\hat{v} = v \circ Q_\kappa \circ F_\kappa$.

Remark 5.1 *We note that a similar construction of the element mappings for general meshes consisting of curved quadrilateral elements has also been employed for both shape-regular and anisotropic meshes in the articles [73] and [42], respectively. The key difference in the current construction to that proposed in [42] is that here the element mapping F_κ contains information*

about both size and orientation of κ. In contrast, in the construction developed in [42] both orientation and shape information are included in Q_κ, while F_κ only contains information relating to the size of κ.

Remark 5.2 *Within this construction we admit meshes with possibly hanging nodes; for simplicity, we shall suppose that the mesh $\mathcal{T}_h$ is 1-irregular, cf. [73].*

Associated with $\mathcal{T}_h$, we introduce the broken Sobolev space of order $s \geq 0$ defined by

$$H^s(\Omega, \mathcal{T}_h) = \{u \in L_2(\Omega) : \ u|_\kappa \in H^s(\kappa) \quad \forall \kappa \in \mathcal{T}_h\} \ ,$$

equipped with the broken Sobolev norm and seminorm, denoted, respectively, by

$$\|u\|_{s,\mathcal{T}_h} = \left(\sum_{\kappa \in \mathcal{T}_h} \|u\|^2_{H^s(\kappa)} \right)^{\frac{1}{2}}, \quad |u|_{s,\mathcal{T}_h} = \left(\sum_{\kappa \in \mathcal{T}_h} |u|^2_{H^s(\kappa)} \right)^{\frac{1}{2}}.$$

For $u \in H^1(\Omega, \mathcal{T}_h)$ we define the broken gradient $\nabla_{\mathcal{T}_h} u$ of u by $(\nabla_{\mathcal{T}_h} u)|_\kappa = \nabla(u|_\kappa)$, $\kappa \in \mathcal{T}_h$.

5.3 Interior penalty discontinuous Galerkin method

We introduce the (symmetric) interior penalty DG discretization of the advection–diffusion–reaction problem (110), (112), cf. Section 3. For ease of presentation, we recall the following notation. Given a polynomial degree $p \geq 1$ we define the finite element space $V_{h,p}$ as follows

$$V_{h,p} \ = \ \{u \in L_2(\Omega) \ : \ u|_\kappa \circ Q_\kappa \circ F_\kappa \in R_p(\hat{\kappa}); \ \kappa \in \mathcal{T}_h\} \, ,$$

where R_p is $\mathcal{P}_p$, when $\hat{\kappa}$ is the unit d–simplex, or R_p is $\mathcal{Q}_p$, when $\hat{\kappa} = (-1,1)^d$. Here, $\mathcal{P}_p$ denotes the set of polynomials of total degree p on $\hat{\kappa}$ and $\mathcal{Q}_p(\hat{\kappa})$, the set of all tensor-product polynomials on $\hat{\kappa}$ of degree p in each coordinate direction.

An *interior face* of $\mathcal{T}_h$ is defined as the (non-empty) $(d-1)$–dimensional interior of $\partial\kappa^+ \cap \partial\kappa^-$, where κ^+ and κ^- are two adjacent elements of $\mathcal{T}_h$, not necessarily matching. A *boundary face* of $\mathcal{T}_h$ is defined as the (non-empty) $(d-1)$–dimensional interior of $\partial\kappa \cap \Gamma$, where κ is a boundary element of $\mathcal{T}_h$. We denote by $\Gamma_{\mathcal{I}}$ the union of all interior faces of $\mathcal{T}_h$. Let κ^+ and κ^- be two adjacent elements of $\mathcal{T}_h$, and $\mathbf{x}$ an arbitrary point on the interior face $f = \partial\kappa^+ \cap \partial\kappa^-$. Furthermore, let v and $\mathbf{q}$ be scalar- and vector-valued functions, respectively, that are smooth inside each element $\kappa^\pm$. By $(v^\pm, \mathbf{q}^\pm)$, we denote the traces of $(v, \mathbf{q})$ on f taken from within the interior of $\kappa^\pm$, respectively. Then, the averages of v and $\mathbf{q}$ at $\mathbf{x} \in f$ are given by

$$\{\!\{v\}\!\} = \frac{1}{2}(v^+ + v^-), \qquad \{\!\{\mathbf{q}\}\!\} = \frac{1}{2}(\mathbf{q}^+ + \mathbf{q}^-),$$

respectively. Similarly, the jumps of v and $\mathbf{q}$ at $\mathbf{x} \in \kappa$ are given by

$$[\![v]\!] = v^+ \, \mathbf{n}_{\kappa^+} + v^- \, \mathbf{n}_{\kappa^-}, \qquad [\![\mathbf{q}]\!] = \mathbf{q}^+ \cdot \mathbf{n}_{\kappa^+} + \mathbf{q}^- \cdot \mathbf{n}_{\kappa^-},$$

respectively, where we denote by $\mathbf{n}_{\kappa^\pm}$ the unit outward normal vector of $\kappa^\pm$, respectively. On a boundary face $f \subset \Gamma$, we set $\{\!\{v\}\!\} = v$, $\{\!\{\mathbf{q}\}\!\} = \mathbf{q}$, and $[\![v]\!] = v\mathbf{n}$, where $\mathbf{n}$ denotes the unit outward normal vector on the boundary Γ.

Given that κ is an element in the subdivision $\mathcal{T}_h$, we denote by $\partial\kappa$ the union of $(d-1)$–dimensional open faces of κ. Let $\mathbf{x} \in \partial\kappa$ and suppose that $\mathbf{n}_\kappa(\mathbf{x})$ denotes the unit outward normal vector to $\partial\kappa$ at $\mathbf{x}$. With these conventions, we define the inflow and outflow parts of $\partial\kappa$, respectively, by

$$\partial_-\kappa = \{\mathbf{x} \in \partial\kappa : \ \mathbf{b}(\mathbf{x}) \cdot \mathbf{n}_\kappa(\mathbf{x}) < 0\} \ , \quad \partial_+\kappa = \{\mathbf{x} \in \partial\kappa : \ \mathbf{b}(\mathbf{x}) \cdot \mathbf{n}_\kappa(\mathbf{x}) \geq 0\} \ .$$

For simplicity of presentation, we suppose that the entries of the matrix a are constant on each element κ in $\mathcal{T}_h$; i.e.,

$$a \in [V_{h,0}]^{d\times d}_{\rm sym} . \tag{117}$$

We note that, with minor changes only, our results can easily be extended to the case of $\sqrt{a} \in [V_{h,q}]^{d\times d}_{\rm sym}$, $q \geq 0$; moreover, for general $a \in L_\infty(\Omega)^{d\times d}_{\rm sym}$, the analysis proceeds in a similar manner, based on employing the modified DG method proposed in [46]. In the following, we write $\bar{a} = |\sqrt{a}\,|_2^2$, where $|\cdot|_2$ denotes the matrix norm subordinate to the l_2–vector norm on $\mathbb{R}^d$ and $\bar{a}_\kappa = \bar{a}|_\kappa$.

The interior penalty DG approximation of (110), (112) is defined as follows: find u_h in $V_{h,p}$ such that

$$\mathcal{B}(u_h, v) = \ell(v) \tag{118}$$

for all $v \in V_{h,p}$. Here, the bilinear form $\mathcal{B}(\cdot,\cdot)$ is defined by

$$\mathcal{B}(w,v) = \mathcal{B}_a(w,v) + \mathcal{B}_{\mathbf{b}}(w,v) - \mathcal{B}_f(v,w) - \mathcal{B}_f(w,v) + \mathcal{B}_\vartheta(w,v) \, ,$$

where

$$\begin{aligned}
\mathcal{B}_a(w,v) &= \sum_{\kappa\in\mathcal{T}_h} \int_\kappa a\nabla w \cdot \nabla v \, {\rm d}\mathbf{x} \, , \\
\mathcal{B}_{\mathbf{b}}(w,v) &= \sum_{\kappa\in\mathcal{T}_h} \Big\{ - \int_\kappa (w\,\mathbf{b}\cdot\nabla v - cwv) \, {\rm d}\mathbf{x} \\
&\qquad + \int_{\partial_+\kappa} (\mathbf{b}\cdot\mathbf{n}_\kappa)\, w^+ v^+ \, {\rm d}s + \int_{\partial_-\kappa\setminus\Gamma} (\mathbf{b}\cdot\mathbf{n}_\kappa)\, w^- v^+ \, {\rm d}s \Big\} \, , \\
\mathcal{B}_f(w,v) &= \int_{\Gamma_{\mathcal{I}}\cup\Gamma_{\rm D}} \{\!\{ a\nabla_h w \}\!\} \cdot [\![v]\!] \, {\rm d}s \, , \qquad \mathcal{B}_\vartheta(w,v) = \int_{\Gamma_{\mathcal{I}}\cup\Gamma_{\rm D}} \vartheta [\![w]\!] \cdot [\![v]\!] \, {\rm d}s \, ,
\end{aligned}$$

and the linear functional $\ell(\cdot)$ is given by

$$\begin{aligned}
\ell(v) &= \sum_{\kappa\in\mathcal{T}_h} \Big(\int_\kappa f v \, {\rm d}\mathbf{x} - \int_{\partial_-\kappa\cap(\Gamma_{\rm D}\cup\Gamma_-)} (\mathbf{b}\cdot\mathbf{n}_\kappa)\, g_{\rm D}\, v^+ \, {\rm d}s \\
&- \int_{\partial\kappa\cap\Gamma_{\rm D}} g_{\rm D}((a\nabla v^+)\cdot\mathbf{n}_\kappa) \, {\rm d}s + \int_{\partial\kappa\cap\Gamma_{\rm N}} g_{\rm N} v^+ \, {\rm d}s + \int_{\partial\kappa\cap\Gamma_{\rm D}} \vartheta g_{\rm D} v^+ \, {\rm d}s \Big) \, .
\end{aligned}$$

Here ϑ is called the *discontinuity-penalization* parameter and is defined by $\vartheta|_f = \vartheta_f$ for $f \subset \Gamma_{\mathcal{I}} \cup \Gamma_{\rm D}$, where ϑ_f is a nonnegative constant on face f. The precise choice of ϑ_f, which

depends on a and the discretization parameters, will be discussed in detail in the next section. We shall adopt the convention that faces $f \subset \Gamma_{\mathcal{I}} \cup \Gamma_{\mathrm{D}}$ with $\vartheta|_f = 0$ are omitted from the integrals appearing in the definition of $\mathcal{B}_\vartheta(w,v)$ and $\ell(v)$, although we shall not highlight this explicitly in our notation; the same convention is adopted in the case of integrals where the integrand contains the factor $1/\vartheta$. Thus, in particular, the definition of the DG-norm, cf. (119) below, is meaningful even if $\vartheta|_f$ happens to be equal to zero on certain faces $f \subset \Gamma_{\mathcal{I}} \cup \Gamma_{\mathrm{D}}$, given that such faces are understood to be excluded from the region of integration. For details concerning the construction of the DG method (118), we refer the reader to the article [50], for example.

Remark 5.3 *For notational simplicity, we have neglected the superscript '+' on the elemental domain $\kappa \in \mathcal{T}_h$ in the definition of both the bilinear form $\mathcal{B}(\cdot,\cdot)$ and the linear functional $\ell(\cdot)$. With this in mind, on an interior face $f \subset \partial\kappa \cap \partial\kappa^-$, where κ and κ^- are two adjacent elements of $\mathcal{T}_h$, the notation $v^\pm$ is used denote the traces of v on f taken from within the interior of κ and κ^-, respectively.*

5.4 Stability analysis

Before embarking on the error analysis of the DG method (118), we first derive some preliminary results. Let us first introduce the DG–norm $|||\cdot|||$ by

$$
\begin{aligned}
|||w|||^2 \;=\; & \sum_{\kappa\in\mathcal{T}_h} \Big(\|\sqrt{a}\,\nabla w\|^2_{L_2(\kappa)} + \|c_0 w\|^2_{L_2(\kappa)} + \frac{1}{2}\|w^+\|^2_{\partial_-\kappa\cap(\Gamma_{\mathrm{D}}\cup\Gamma_-)} \\
& \quad + \frac{1}{2}\|w^+ - w^-\|^2_{\partial_-\kappa\setminus\Gamma} + \frac{1}{2}\|w^+\|^2_{\partial_+\kappa\cap\Gamma} \Big) \\
& \quad + \int_{\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}} \vartheta |[\![w]\!]|^2 \,\mathrm{d}s + \int_{\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}} \frac{1}{\vartheta} |\{\!\{a\,\nabla w\}\!\}|^2 \,\mathrm{d}s \,, \qquad (119)
\end{aligned}
$$

where $\|\cdot\|_\tau$, $\tau \subset \partial\kappa$, denotes the (semi)norm associated with the (semi)inner-product $(v,w)_\tau = \int_\tau |\mathbf{b}\cdot\mathbf{n}_\kappa| vw \,\mathrm{d}s$, and c_0 is as defined in (114). We remark that the above definition of $|||\cdot|||$ represents a slight modification of the norm considered in [74]; in the case $\mathbf{b} \equiv \mathbf{0}$, (119) corresponds to the norm proposed by Baumann *et al.* [18, 91] and Baker *et al.* [9], cf. [95].

For a given face $f \subset \Gamma_{\mathcal{I}} \cup \Gamma_{\mathrm{D}}$, such that $f \subset \partial\kappa$, for some $\kappa \in \mathcal{T}_h$, we write $\tilde{f}$ and $\hat{f}$ to denote the respective faces of the mapped elements $\tilde{\kappa}$ and $\hat{\kappa}$, respectively, based on employing the element mappings Q_κ and F_κ. More precisely, we write $\tilde{f} = Q_\kappa^{-1}(f)$ and $\hat{f} = F_\kappa^{-1}(\tilde{f})$. Further, we define m_f, $m_{\tilde{f}}$, and $m_{\hat{f}}$ to denote the $(d-1)$–dimensional measure (volume) of the faces f, $\tilde{f}$, and $\hat{f}$, respectively; clearly, in two–dimensions, i.e., $d = 2$, $m_{\hat{f}}$, the length of the corresponding face on the canonical element, is equal to 2 when quadrilateral elements are employed. In view of (115), we note that there exists a positive constant C_5, such that

$$
C_5^{-1} m_{\tilde{f}} \le m_f \le C_5 m_{\tilde{f}} \qquad (120)
$$

for every face $f \subset \Gamma_{\mathcal{I}} \cup \Gamma_{\mathrm{D}}$. Moreover, the surface Jacobian $S_{f,\tilde{f}}$ arising in the transformation of the face f to $\tilde{f}$ may be uniformly bounded in the following manner

$$
\|S_{f,\tilde{f}}\|_{L_\infty(\tilde{f})} \le C_6 \qquad (121)
$$

for all faces $f \subset \Gamma_{\mathcal{I}} \cup \Gamma_{\mathrm{D}}$, where C_6 is a positive constant.

Let us now quote the following inverse inequality.

Lemma 5.4 *Let κ be an element contained in the mesh $\mathcal{T}_h$ and let f denote one of its faces. Then, the following inverse inequality holds*

$$\|v\|^2_{L_2(f)} \leq C_{\mathrm{inv}} \frac{m_f}{m_\kappa} \|v\|^2_{L_2(\kappa)} \tag{122}$$

for all v such that $v \circ Q_\kappa \circ F_\kappa \in R_p(\hat{\kappa})$, where C_{inv} is a constant which depends only on the dimension d and the polynomial degree p.

Proof: On the reference element $\hat{\kappa}$, for any function $\hat{v} \in R_p(\hat{\kappa})$, there exists a positive constant C'_{inv}, such that

$$\|\hat{v}\|^2_{L_2(\hat{f})} \leq C'_{\mathrm{inv}} \|\hat{v}\|^2_{L_2(\hat{\kappa})}; \tag{123}$$

see, for example, [7]. Thereby, employing (121) and (120) we deduce that

$$\|v\|^2_{L_2(f)} \leq C_6 \|\tilde{v}\|^2_{L_2(\tilde{f})} = C_6 \frac{m_{\tilde{f}}}{m_{\hat{f}}} \|\hat{v}\|^2_{L_2(\hat{f})} \leq \frac{C_6}{C_5} \frac{m_f}{m_{\hat{f}}} \|\hat{v}\|^2_{L_2(\hat{f})}. \tag{124}$$

In an analogous manner, by exploiting (116) and (115) gives

$$\|\hat{v}\|^2_{L_2(\hat{\kappa})} = \det(F_\kappa^{-1}) \|\tilde{v}\|^2_{L_2(\tilde{\kappa})} = \frac{m_{\hat{\kappa}}}{m_{\tilde{\kappa}}} \|\tilde{v}\|^2_{L_2(\tilde{\kappa})} \leq C_4 \frac{m_{\hat{\kappa}}}{m_\kappa} \|\tilde{v}\|^2_{L_2(\tilde{\kappa})} \leq C_1 C_4 \frac{m_{\hat{\kappa}}}{m_\kappa} \|v\|^2_{L_2(\kappa)}. \tag{125}$$

Inserting (124) and (125) into (123) gives the desired result. □

Remark 5.5 *The inverse inequality stated in Lemma 5.4 is an extension of the standard result employed on isotropic finite element meshes to the case when anisotropic elements may be present. Indeed, in the isotropic setting, we have that $m_\kappa \approx h_\kappa^d$ and $m_f \approx h_\kappa^{d-1}$, where h_κ denotes the diameter of the element $\kappa \in \mathcal{T}_h$; thereby, the scaling on the right–hand side of the inequality (122) is of size $1/h_\kappa$, as expected. Moreover, this result extends the inverse inequality stated in [42] to the case when the affine mapping F_κ includes not only size, but also orientation information, cf. above.*

We now define the function $\mathtt{h}$ in $L_\infty(\Gamma_{\mathcal{I}} \cup \Gamma_{\mathrm{D}})$, as $\mathtt{h}(\mathbf{x}) = \min\{m_{\kappa_1}, m_{\kappa_2}\}/m_f$, if $\mathbf{x}$ is in the interior of $f = \partial\kappa_1 \cap \partial\kappa_2$ for two neighboring elements in the mesh $\mathcal{T}_h$, and $\mathtt{h}(\mathbf{x}) = m_\kappa/m_f$, if $\mathbf{x}$ is in the interior of $f = \partial\kappa \cap \Gamma_{\mathrm{D}}$. We note that in the isotropic setting we observe that $\mathtt{h} \sim h$, where h denotes the mesh local mesh size, cf. Remark 5.5 above. Similarly, we define the function $\mathtt{a}$ in $L_\infty(\Gamma_{\mathcal{I}} \cup \Gamma_{\mathrm{D}})$ by $\mathtt{a}(\mathbf{x}) = \max\{\bar{a}_{\kappa_1}, \bar{a}_{\kappa_2}\}$ if $\mathbf{x}$ is in the interior of $f = \partial\kappa_1 \cap \partial\kappa_2$, and $\mathtt{a}(\mathbf{x}) = \bar{a}_\kappa$ if $\mathbf{x}$ is in the interior of $\partial\kappa \cap \Gamma_{\mathrm{D}}$. With this notation, we now provide the following coercivity result for the bilinear form $\mathcal{B}(\cdot,\cdot)$ over $V_{h,p} \times V_{h,p}$.

Theorem 5.6 *Define the discontinuity-penalization parameter ϑ arising in (118) by*

$$\vartheta|_f \equiv \vartheta_f = C_\vartheta \frac{\mathtt{a}}{\mathtt{h}} \quad \textit{for} \quad f \subset \Gamma_{\mathcal{I}} \cup \Gamma_{\mathrm{D}}, \tag{126}$$

where C_ϑ is a sufficiently large positive constant (see Remark 5.7 below). Then, there exists a positive constant C, which depends only on the dimension d and the polynomial degree p, such that

$$\mathcal{B}(v,v) \geq C |||v|||^2 \quad \forall v \in V_{h,p}. \tag{127}$$

Proof: This result follows by application of the inverse estimate derived in Lemma 5.4, following the general argument presented by Prudhomme *et al.* [95] in the case when $\mathbf{b} \equiv \mathbf{0}$; cf., also [74]. □

Remark 5.7 *Theorem 5.6 indicates that the DG scheme is coercive over $V_{h,p} \times V_{h,p}$ provided that the constant $C_\vartheta > 0$ arising in the definition of the discontinuity–penalization parameter ϑ, is chosen sufficiently large. More precisely, C_ϑ should be selected to be a positive constant which is greater than $C_f C_{\rm inv}/2$, where $C_{\rm inv}$ is the constant arising in the inverse inequality stated in Lemma 5.4 and*

$$C_f = \max_{\kappa \in \mathcal{T}_h} card\{f \subset \Gamma_{\mathcal{I}} \cup \Gamma_{\rm D} : f \subset \partial\kappa\};$$

the restriction to 1–irregular meshes ensures that C_f is uniformly bounded independently of the mesh size.

For the proceeding error analysis, we assume that the solution u to the boundary value problem (110), (112) is sufficiently smooth: namely, $u \in H^{3/2+\varepsilon}(\Omega, \mathcal{T}_h)$, $\varepsilon > 0$, and the functions u and $(a\nabla u) \cdot \mathbf{n}_f$ are continuous across each face $f \subset \partial\kappa \backslash \Gamma$ that intersects the subdomain of ellipticity, $\Omega_a = \{\mathbf{x} \in \bar{\Omega} : \boldsymbol{\zeta}^\top a(\mathbf{x})\boldsymbol{\zeta} > 0 \ \ \forall \boldsymbol{\zeta} \in \mathbb{R}^d\}$. If this smoothness requirement is violated, the discretization method has to be modified accordingly, cf. [74]. We note that under these assumptions, the following Galerkin orthogonality property holds:

$$\mathcal{B}(u - u_h, v) = 0 \quad \forall v \in V_{h,p}\,. \tag{128}$$

For simplicity of presentation, it will be assumed in the proceeding analysis, as well as in Section 5.6, that the velocity vector $\mathbf{b}$ satisfies the following assumption:

$$\mathbf{b} \cdot \nabla_{\mathcal{T}_h} v \in V_{h,p} \quad \forall v \in V_{h,p}\,. \tag{129}$$

To ensure that (110) is then meaningful (i.e., that the characteristic curves of the differential operator L are correctly defined), we still assume that $\mathbf{b} \in \left[W^1_\infty(\Omega)\right]^d$.

Remark 5.8 *We note that hypothesis (129) is a standard condition assumed for the analysis of the hp–version of the DG method; see, for example, [42, 50, 74]. Indeed, this condition is essential for the derivation of* a priori *error bounds which are optimal in both the mesh size h and spectral order p; in the absence of this assumption, optimal h–convergence bounds may still be derived, though a loss of $p^{3/2}$ is observed in the resulting error analysis, unless the scheme (118) is supplemented by appropriate streamline–diffusion stabilization, cf. the discussion in [73]. Given that within the current setting, we are only interested in deriving error bounds for the h–version of the DG method, hypothesis (129) is indeed unnecessary, but for simplicity of presentation, we retain this assumption.*

5.5 Approximation results

In this section we develop the necessary approximation results needed for the forthcoming *a priori* error estimation developed in Section 5.6. To this end, on the reference element $\hat{\kappa}$, we

define $\hat{\Pi}_p$ to denote the orthogonal projector in $L_2(\hat{\kappa})$ onto the space of polynomials $R_p(\hat{\kappa})$; i.e., given that $\hat{v} \in L_2(\hat{\kappa})$, we define $\hat{\Pi}_p \hat{v}$ by

$$(\hat{v} - \hat{\Pi}_p \hat{v}, \hat{w})_{\hat{\kappa}} = 0$$

for all $\hat{w} \in R_p(\hat{\kappa})$, where $(\cdot, \cdot)_{\hat{\kappa}}$ denotes the $L_2(\hat{\kappa})$ inner product. Similarly, we define the L_2-projection operators $\tilde{\Pi}_p$ and Π_p on $\tilde{\kappa}$ and κ, respectively, by the relations

$$\tilde{\Pi}_p \tilde{v} := (\hat{\Pi}_p(\tilde{v} \circ F_\kappa)) \circ F_\kappa^{-1}, \qquad \Pi_p v := (\tilde{\Pi}_p(v \circ Q_\kappa)) \circ Q_\kappa^{-1},$$

for $\tilde{v} \in L_2(\tilde{\kappa})$ and $v \in L_2(\kappa)$, respectively. Also, we define the (element-wise) L_2–projection operator onto $V_{h,p}$ by $(\Pi_p^{\mathcal{T}_h} u)|_\kappa := \Pi_p(u|_\kappa)$ for all elements $\kappa \in \mathcal{T}_h$.

We remark that this choice of projector is essential in the following *a priori* error analysis, in order to ensure that

$$(u - \Pi_p^{\mathcal{T}_h} u, \mathbf{b} \cdot \nabla_{\mathcal{T}_h} v) = 0 \tag{130}$$

for all v in $V_{h,p}$, cf. the proofs of Lemma 5.22 and Theorem 5.23 below. We remark that this same choice of projector is also necessary in the corresponding case when (129) fails to hold; in this situation an equality of the form (130) with $\mathbf{b}$ replaced by a suitable projection of $\mathbf{b}$ is still necessary for the underlying analysis; cf. [41], Chapter 5.

With this notation, we now quote the following approximation results on the reference element $\hat{\kappa}$.

Lemma 5.9 *Let $\hat{\kappa}$ be the reference element, and let $\hat{f}$ denote one of its faces. Given a function $\hat{v} \in H^k(\hat{\kappa})$, the following error bounds hold for $m = 0, 1$:*

$$|\hat{v} - \hat{\Pi}_p \hat{v}|_{H^m(\hat{\kappa})} \leq C|\hat{v}|_{H^s(\hat{\kappa})}, \qquad m \leq s \leq \min(p+1, k), \tag{131}$$

$$|\hat{v} - \hat{\Pi}_p \hat{v}|_{H^m(\hat{f})} \leq C|\hat{v}|_{H^s(\hat{\kappa})}, \qquad m+1 \leq s \leq \min(p+1, k), \tag{132}$$

where C is a positive constant which depends only on the dimension d and the polynomial order p.

Proof: The proof of (131) is standard; see [30], for example. The approximation result (132) follows upon application of the multiplicative trace inequality, cf. [73]. □

Corollary 5.10 *Using the notation of Lemma 5.9, there exists a positive constant C, which depends only on the dimension d and the polynomial order p, such that for $m = 0, 1$:*

$$|v - \Pi_p v|_{H^m(\kappa)} \leq C|\det(J_{F_\kappa})|^{1/2} \|J_{F_\kappa}^{-\top}\|_2^m \, |\hat{v}|_{H^s(\hat{\kappa})}, \quad m \leq s \leq \min(p+1, k), \tag{133}$$

$$|v - \Pi_p v|_{H^m(f)} \leq C|m_f|^{1/2} \|J_{F_\kappa}^{-\top}\|_2^m \, |\hat{v}|_{H^s(\hat{\kappa})}, \qquad m+1 \leq s \leq \min(p+1, k). \tag{134}$$

Proof: The proof of the each inequality stated in the corollary is based on exploiting a standard scaling argument to the respective left–hand sides of the approximation results stated in Lemma 5.9, together with (115), (120), (121), and (124). Indeed, the proof of (133) exploits (131), together with the following (scaling) inequality

$$\begin{aligned} |v - \Pi_p v|^2_{H^m(\kappa)} &\leq \|\det J_{Q_\kappa}\|_{L_\infty(\kappa)} \|J_{Q_\kappa}^{-\top}\|^{2m}_{L_\infty(\kappa)} \, |\tilde{v} - \tilde{\Pi}_p \tilde{v}|^2_{H^m(\tilde{\kappa})} \\ &\leq C_1 (C_2)^{2m} |\tilde{v} - \tilde{\Pi}_p \tilde{v}|^2_{H^m(\tilde{\kappa})} \leq C_1 (C_2)^{2m} |\det J_{F_\kappa}| \, \|J_{F_\kappa}^{-\top}\|_2^{2m} \, |\hat{v} - \hat{\Pi}_p \hat{v}|^2_{H^m(\hat{\kappa})}; \end{aligned} \tag{135}$$

here we have used (115). Finally, employing (115), (121), and (120), we deduce that

$$|v - \Pi_p v|^2_{H^m(f)} \leq C_3^m C_6 |\tilde{v} - \tilde{\Pi}_p \tilde{v}|^2_{H^m(\tilde{f})} \leq \frac{C_3^m C_6}{C_5} \frac{m_f}{m_{\hat{f}}} \|J_{F_\kappa}^{-\top}\|_2^{2m} |\hat{v} - \hat{\Pi}_p \hat{v}|^2_{H^m(\hat{f})}. \tag{136}$$

Upon substituting (136) into (132), we deduce (134). □

Finally, it remains to scale the $H^s(\hat{\kappa})$, $s \geq 0$, semi-norm defined on the reference element $\hat{\kappa}$ to $\tilde{\kappa}$ based on employing the affine element transformation F_κ. In order to retain the anisotropic mesh information within the Jacobian J_{F_κ}, we first re-write the square of the $H^s(\hat{\kappa})$ semi-norm of a function $\hat{v}$ in terms of the integral of the square of the Frobenius norm of an sth–order tensor containing the s–order derivatives of $\hat{v}$. With this definition the transformation of the s–order derivatives of $\hat{v}$ defined over $\hat{\kappa}$ may naturally be transformed to derivatives of the (mapped) function $\tilde{v}$ defined over $\tilde{\kappa}$. Indeed, for the case when $s = 2$, this approach is analogous to the technique employed in [40].

To this end, we now introduce the following tensor notation; here, and in the following we use calligraphic letters $\mathcal{A}, \mathcal{B}, \ldots$ to denote Nth–order tensors, where it is understood that a 0th–order tensor is a scalar, a 1st–order tensor is a vector, a 2nd–order tensor is a matrix, and so on. The following discussion regarding tensors is based on the work presented in the article [84].

Definition 5.11 *For an Nth–order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, the matrix unfolding $A_{(n)} \in \mathbb{R}^{I_n \times (I_{n+1} I_{n+2} \ldots I_N I_1 I_2 \ldots I_{n-1})}$, $n = 1, \ldots, N$, contains the element $a_{i_1 i_2 \ldots i_N}$ at the position with row number i_n and column number equal to*

$$\begin{aligned}&(i_{n+1} - 1) I_{n+2} I_{n+3} \ldots I_N I_2 \ldots I_{n-1} + (i_{n+2} - 1) I_{n+3} I_{n+4} \ldots I_N I_1 I_2 \ldots I_{n-1} + \ldots \\ &+ (i_N - 1) I_1 I_2 \ldots I_{n-1} + (i_1 - 1) I_2 I_3 \ldots I_{n-1} + (i_2 - 1) I_3 I_4 \ldots I_{n-1} + \ldots + i_{n-1}.\end{aligned}$$

In essence a matrix unfolding represents a splitting of an Nth–order tensor into a vector of $(N-1)$th–order tensors. These $(N-1)$th–order tensors are then recursively unfolded until 2nd–order tensors (matrices) are realised. Figure 28 shows the three unfoldings possible for a $3rd$–order tensor.

This definition prompts us to consider a way of multiplying a tensor by a matrix. Clearly if we have a matrix $U \in \mathbb{R}^{J_n \times I_n}$ then we can pre-multiply $A_{(n)}$ by U. Forming an Nth–order tensor from $UA_{(n)}$ by reversing the matrix unfolding procedure we have the product of a matrix and a tensor, giving rise to a tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots I_N}$. Diagrammatically this can be represented as

$$\underbrace{\mathcal{A} \xrightarrow{\text{Unfold}} A_{(n)} \xrightarrow{U\times} UA_{(n)} \xrightarrow{\text{Refold}} \mathcal{A} \times_n U}_{\times_n U}.$$

We formalize this in the following definition.

Definition 5.12 *The n-mode product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ by a matrix $U \in \mathbb{R}^{J_n \times I_n}$, denoted by $\mathcal{A} \times_n U$, is an $I_1 \times I_2 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots I_N$-tensor of which the entries are given by*

$$(\mathcal{A} \times_n U)_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} := \sum_{i_n = 1}^{I_n} (\mathcal{A})_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} (U)_{j_n i_n}.$$

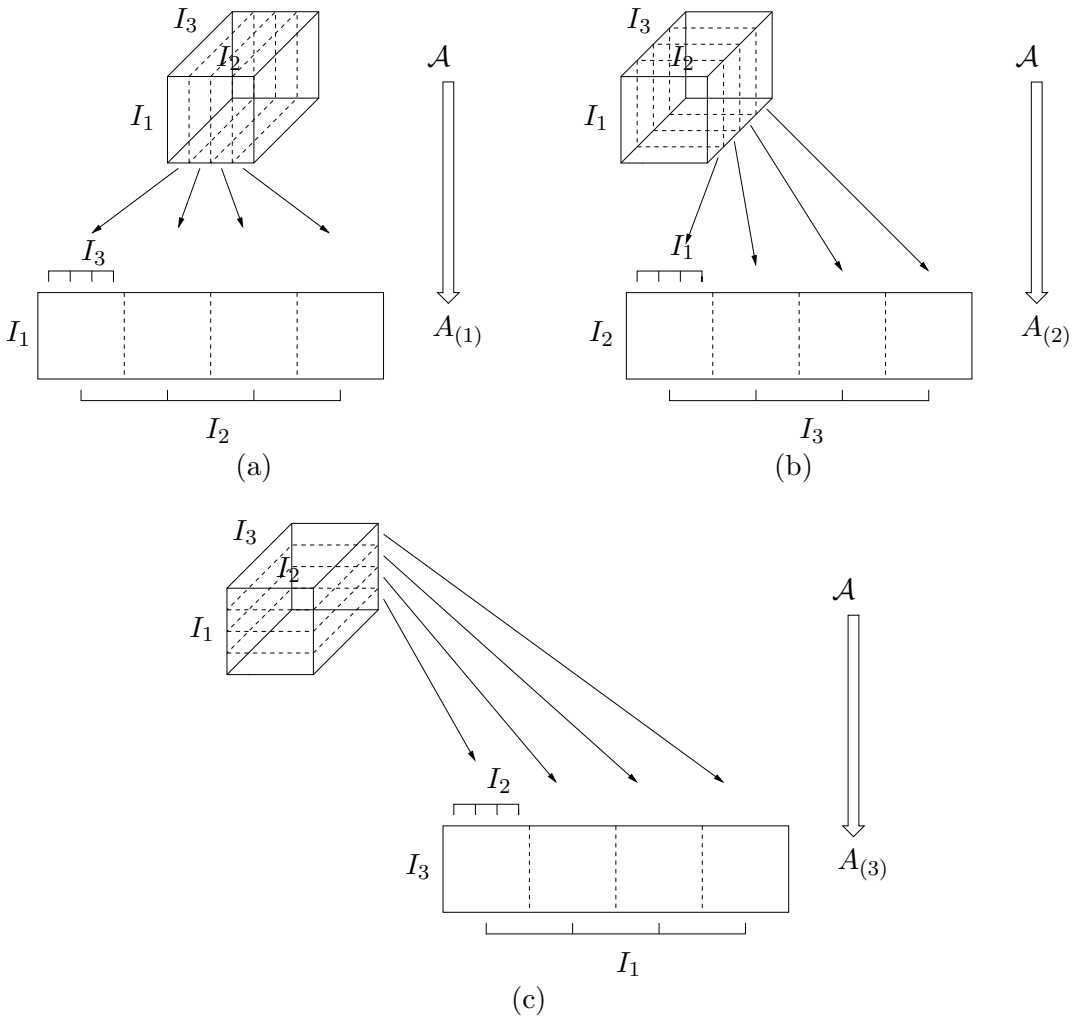


Figure 28: Matrix unfolding of a 3rd order tensor: (a) First unfolding; (b) Second unfolding; (c) Third unfolding.

Lemma 5.13 *For $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and $U \in \mathbb{R}^{J_n \times I_n}$, we have that*

$$(\mathcal{A} \times_n U)_{(n)} = U\mathcal{A}_{(n)}.$$

Proof: Consider element $(\mathcal{A} \times_n U)_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N}$, its position in $(\mathcal{A} \times_n U)_{(n)}$ is at row number j_n and column number k, where

$$\begin{aligned} k &= (i_{n+1}-1)I_{n+2}I_{n+3}\ldots I_N I_2 \ldots I_{n-1} + (i_{n+2}-1)I_{n+3}I_{n+4}\ldots I_N I_1 I_2 \ldots I_{n-1} + \ldots \\ &\quad + (i_N - 1)I_1 I_2 \ldots I_{n-1} + (i_1 - 1)I_2 I_3 \ldots I_{n-1} + (i_2 - 1)I_3 I_4 \ldots I_{n-1} + \ldots + i_{n-1}. \end{aligned}$$

Now,

$$(U\mathcal{A}_{(n)})_{j_n k} = \sum_{i_n=1}^{I_n} (U)_{j_n i_n} (\mathcal{A}_{(n)})_{i_n k} = \sum_{i_n}^{I_n} (\mathcal{A})_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} (U)_{j_n i_n}.$$

Hence, $(\mathcal{A} \times_n U)_{(n)} = U\mathcal{A}_{(n)}$, as required. □

By considering a vector $\mathbf{v}$ as an $I_n \times 1$ matrix, then an n-mode product of $\mathbf{v}^\top$ and $\mathcal{A}$ can be formed to produce an $I_1 \times I_2 \times \ldots \times I_{n-1} \times 1 \times I_{n+1} \times \ldots \times I_N$-tensor. This tensor could be viewed as an $N-1$-tensor, but instead we leave it as an N-tensor in order that we can form other m-mode products without the value of m having to change. However, if we have a $1 \times 1 \times \ldots \times 1$-tensor then we simply view this as a scalar. The n-mode product satisfies the following property.

Property 1. For a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and the matrices $F \in \mathbb{R}^{J_n \times I_n}$ and $G \in \mathbb{R}^{J_m \times I_m}$, $n \neq m$, we have

$$(\mathcal{A} \times_n F) \times_m G = (\mathcal{A} \times_m G) \times_n F = \mathcal{A} \times_n F \times_m G.$$

We also introduce the Frobenius norm of a tensor.

Definition 5.14 *The Frobenius-norm, $\|\cdot\|_F$, of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is given by*

$$\|\mathcal{A}\|_F^2 = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} (\mathcal{A})_{i_1 i_2 \cdots i_N}^2.$$

Lemma 5.15 *Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and an orthogonal matrix $F \in \mathbb{R}^{I_n \times I_n}$, the following holds*

$$\|\mathcal{A} \times_n F\|_F = \|\mathcal{A}\|_F. \tag{137}$$

Proof: For a matrix $A \in \mathbb{R}^{I_n \times m}$ we have that

$$\|FA\|_F = \|A\|_F. \tag{138}$$

Using the identity in Lemma 5.13, namely, $(\mathcal{A} \times_n F)_{(n)} = F\mathcal{A}_{(n)}$, we deduce that

$$\|\mathcal{A} \times_n F\|_F = \|F\mathcal{A}_{(n)}\|_F.$$

Given that $\mathcal{A}_{(n)} \in \mathbb{R}^{I_n \times I_{n+1} \ldots I_N \ldots I_1 \ldots I_{n-1}}$, exploiting (138) gives

$$\|\mathcal{A} \times_n F\|_F \;\; = \;\; \|F\mathcal{A}_{(n)}\|_F = \|\mathcal{A}_{(n)}\|_F = \|\mathcal{A}\|_F.$$

□

In order to rescale $|\hat{v}|_{H^s(\hat{\kappa})}$ to the corresponding quantity on $\tilde{\kappa}$, we first note that

$$|\hat{v}|_{H^s(\hat{\kappa})}^2 = \int_{\hat{\kappa}} \|\hat{\mathcal{D}}^s(\hat{v})\|_F^2 \, \mathrm{d}\hat{\mathbf{x}},$$

where $\hat{\mathcal{D}}^s(\hat{v}) \in \mathbb{R}^{d \times d \times \cdots \times d}$ is the sth–order tensor containing the sth–order derivatives of $\hat{v}$ with respect to the coordinate system $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_d)$, i.e.,

$$(\hat{\mathcal{D}}^s(\hat{v}))_{i_1, i_2, \ldots, i_s} = \frac{\partial^s \hat{v}}{\partial \hat{x}_{i_1} \cdots \partial \hat{x}_{i_s}}, \quad i_k = 1, \ldots, d, \text{ for } k = 1, \ldots, s.$$

Thereby, for $s = 0$, $\hat{\mathcal{D}}^s(\hat{v}) = \hat{v}$, for $s = 1$, $\hat{\mathcal{D}}^s(\hat{v})$ is the gradient vector, and for $s = 2$, $\hat{\mathcal{D}}^s(\hat{v})$ is the Hessian matrix of second–order derivatives. Writing $\tilde{\mathcal{D}}^s(\tilde{v}) \in \mathbb{R}^{d \times d \times \cdots \times d}$ to denote the sth–order tensor containing the sth–order derivatives of $\tilde{v}$ with respect to the coordinate system $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_d)$, we now state the following lemma relating $|\hat{v}|_{H^s(\hat{\kappa})}^2$ to $|\tilde{v}|_{H^s(\tilde{\kappa})}^2$.

Lemma 5.16 *Under the foregoing assumptions, for $\tilde{v} \in H^s(\tilde{\kappa})$, $s \geq 0$, we have that*

$$|\hat{v}|^2_{H^s(\hat{\kappa})} = |\det(J_{F_\kappa}^{-1})| \int_{\tilde{\kappa}} \|\tilde{\mathcal{D}}^s(\tilde{v}) \times_1 J_{F_\kappa}^\top \times_2 J_{F_\kappa}^\top \times_3 \ldots \times_s J_{F_\kappa}^\top\|_F^2 \, \mathrm{d}\tilde{\mathbf{x}}.$$

Proof: The case when $s = 0$ follows trivially. For $s \geq 1$, we first note that the entry $(\hat{\mathcal{D}}^s(\hat{v}))_{i_1 i_2 \ldots i_s}$ may be written in the form

$$\frac{\partial^s \hat{v}}{\partial \hat{x}_{i_1} \cdots \partial \hat{x}_{i_s}} = \sum_{j_1=1}^{d} \cdots \sum_{j_s=1}^{d} (J_{F_\kappa})_{j_1 i_1} \cdots (J_{F_\kappa})_{j_s i_s} \frac{\partial^s \tilde{v}}{\partial \tilde{x}_{j_1} \cdots \partial \tilde{x}_{j_s}},$$

for $i_k = 1, \ldots, d$ and $k = 1, \ldots, s$; this follows by employing an induction argument together with the chain rule. Thereby, from Definition 5.12 and Property 1 above, we deduce that

$$\hat{\mathcal{D}}^s(\hat{v}) = \tilde{\mathcal{D}}^s(\tilde{v}) \times_1 J_{F_\kappa}^\top \times_2 J_{F_\kappa}^\top \times_3 \ldots \times_s J_{F_\kappa}^\top. \tag{139}$$

The statement of the lemma now follows by a simple change of variables. □

Remark 5.17 *For the case when $s = 0$, Lemma 5.16 simply states the change of variable formula for the L_2-norm. For $s = 1$ we note that (139) gives rise to the usual change of variables for the gradient operator, namely,*

$$\hat{\mathcal{D}}^s(\hat{v}) \equiv \nabla_{\hat{\mathbf{x}}} \hat{v} = \tilde{\mathcal{D}}^s(\tilde{v}) \times_1 J_{F_\kappa}^\top = J_{F_\kappa}^\top \nabla_{\tilde{\mathbf{x}}} \tilde{v},$$

where $\nabla_{\hat{\mathbf{x}}}$ and $\nabla_{\tilde{\mathbf{x}}}$ denote the gradient operator with respect to the coordinate systems $\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}}$, respectively. Similarly, for $s = 2$, (139) may be written in the more familiar form $H_{\hat{\mathbf{x}}}(\hat{v}) = J_{F_\kappa}^\top H_{\tilde{\mathbf{x}}}(\tilde{v}) J_{F_\kappa}$, where $H_{\hat{\mathbf{x}}}(\cdot)$ and $H_{\tilde{\mathbf{x}}}(\cdot)$ denote the Hessian matrix operators with respect to the coordinate systems $\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}}$, respectively, cf. [40].

In order to describe the length scales and orientation of the element $\tilde{\kappa}$ we adopt a similar approach to that developed in [40]. To this end, we first need the following definition for the Singular Value Decomposition of a matrix.

Definition 5.18 *A matrix $A \in \mathbb{R}^{m \times n}$ can be decomposed as follows:*

$$A = U \Sigma V^\top,$$

where $U \in \mathbb{R}^{m \times m}$ is an orthogonal matrix termed the left singular matrix, $\Sigma \in \mathbb{R}^{m \times n}$ is a pseudo-diagonal matrix with non-zero entries called the singular values and $V \in \mathbb{R}^{n \times n}$ an orthogonal matrix termed the right singular matrix. This decomposition is called the Singular Value Decomposition (SVD).

It is convention that the singular values σ_i of Σ are non-increasing, that is $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_s \geq 0$, where $s = \min(m, n)$. Figure 29 shows the physical meaning of the SVD for a matrix $A \in \mathbb{R}^{2 \times 2}$, which is assumed to be of full rank. Viewing the matrix A as a map, the left singular matrix $U = [\mathbf{u}_1, \mathbf{u}_2]$ is composed of orthonormal vectors $\mathbf{u}_i$, $i = 1, 2$, which are in the direction of the images of the respective orthonormal vectors $\mathbf{v}_i$ of the matrix $V = [\mathbf{v}_1, \mathbf{v}_2]$. The singular values represent the stretching factors of the corresponding orthonormal vectors;

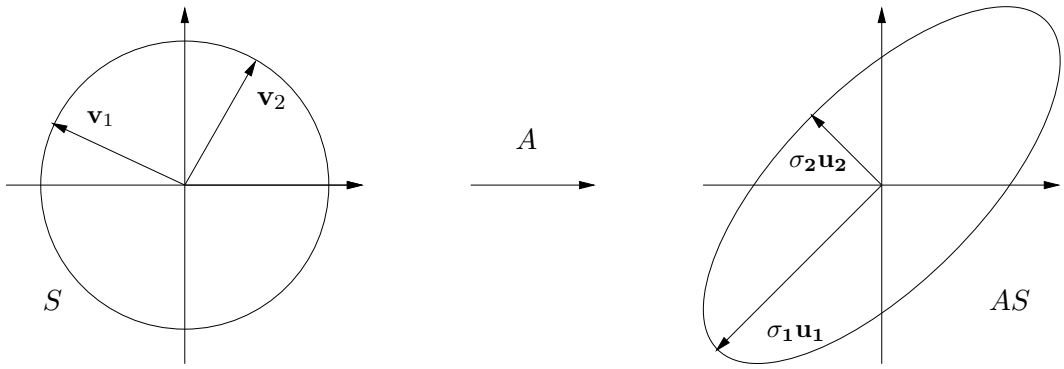


Figure 29: SVD of a 2×2 matrix A.

hence the SVD provides a complete characterisation of the map A. For more information on the Singular Value Decomposition, see, for example, Trefethen & Bau [105].

With this definition, we perform an SVD decomposition of the Jacobi matrix J_{F_κ} of the affine element mapping F_κ. Thereby, we write

$$J_{F_\kappa} = U_\kappa \Sigma_\kappa V_\kappa^\top,$$

where U_κ and V_κ are $d \times d$ orthogonal matrices containing the left and right singular vectors of J_{F_κ}, respectively, and $\Sigma_\kappa = \mathrm{diag}(\sigma_{1,\kappa}, \sigma_{2,\kappa}, \ldots, \sigma_{d,\kappa})$ is a $d \times d$ diagonal matrix containing the singular values $\sigma_{i,\kappa}$, $i = 1, \ldots, d$, of J_{F_κ}. By convention, we assume that $\sigma_{1,\kappa} \geq \sigma_{2,\kappa} \geq \ldots \geq \sigma_{d,\kappa} > 0$. Writing $U_\kappa = (\mathbf{u}_{1,\kappa} \ldots \mathbf{u}_{d,\kappa})$, where $\mathbf{u}_{i,\kappa}$, $i = 1, \ldots, d$, denote the left singular vectors of J_{F_κ}, we note that $\mathbf{u}_{i,\kappa}$, $i = 1, \ldots, d$, give the direction of stretching of the element κ, while $\sigma_{i,\kappa}$, $i = 1, \ldots, d$, give the stretching lengths in the respective directions. Indeed, for axiparallel meshes, as considered in [42], for example, then $\mathbf{u}_{i,\kappa}$, $i = 1, \ldots, d$, will be parallel to the coordinates axes and $\sigma_{i,\kappa}$, $i = 1, \ldots, d$, will denote the local mesh lengths within the respective coordinate direction, cf. Section 6.

With this notation, we make the following observations

$$|\det(J_{F_\kappa})| = \Pi_{i=1}^d \sigma_{i,\kappa}, \quad \|J_{F_\kappa}^{-\top}\|_2 = 1/\sigma_{d,\kappa}, \quad m_f \leq C_7 \, \Pi_{i=1}^{d-1} \sigma_{i,\kappa}, \tag{140}$$

where C_7 is a positive constant independent of the element size. Here, we recall that f is a given face of the element κ. Employing Lemma 5.15, we note that

$$\begin{aligned}
&\|\tilde{\mathcal{D}}^s(\tilde{v}) \times_1 J_{F_\kappa}^\top \times_2 J_{F_\kappa}^\top \times_3 \ldots \times_s J_{F_\kappa}^\top\|_F^2 \\
&\quad = \sum_{i_1=1}^d \sum_{i_2=1}^d \ldots \sum_{i_s=1}^d (\sigma_{i_1,\kappa}\sigma_{i_2,\kappa} \ldots \sigma_{i_s,\kappa})^2 (\tilde{\mathcal{D}}^s(\tilde{v}) \times_1 \mathbf{u}_{i_1,\kappa}^\top \times_2 \mathbf{u}_{i_2,\kappa}^\top \times_3 \ldots \times_s \mathbf{u}_{i_s,\kappa}^\top)^2 \\
&\quad \equiv D_\kappa^s(\tilde{v}, \Sigma_\kappa, U_\kappa).
\end{aligned} \tag{141}$$

Remark 5.19 *We note that, should the mapping F_κ yield a near isotropic element κ, then upon defining the standard isotropic mesh size h_κ by*

$$h_\kappa := \mathrm{diam}(\kappa), \tag{142}$$

we have

$$\sigma_{i,\kappa} \sim h_\kappa, \quad i = 1, \ldots, d.$$

Hence, in this isotropic setting

$$D^s_{\tilde{\kappa}}(\tilde{v}, \Sigma_\kappa, U_\kappa) \sim h_\kappa^{2s}.$$

Thereby, exploiting (140) and (141) together with Corollary 5.10, we deduce the following approximation result.

Theorem 5.20 *Using the notation of Lemma 5.9, there exists a positive constant C, which depends only on the dimension d and the polynomial order p, such that for $m = 0, 1$:*

$$\begin{aligned}
|v - \Pi_p v|_{H^m(\kappa)} &\leq C|\sigma_{d,\kappa}|^{-m} \left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v}, \Sigma_\kappa, U_\kappa)\, \mathrm{d}\tilde{\mathbf{x}}\right]^{1/2}, \quad m \leq s \leq \min(p+1, k), \\
\|v - \Pi_p v\|_{L_2(f)} &\leq C|\sigma_{d,\kappa}|^{-1/2} \left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v}, \Sigma_\kappa, U_\kappa)\, \mathrm{d}\tilde{\mathbf{x}}\right]^{1/2}, \quad 1 \leq s \leq \min(p+1, k), \\
|v - \Pi_p v|_{H^1(f)} &\leq C \left|\frac{m_f}{m_\kappa}\right|^{1/2} |\sigma_{d,\kappa}|^{-1} \left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v}, \Sigma_\kappa, U_\kappa)\, \mathrm{d}\tilde{\mathbf{x}}\right]^{1/2}, \ 2 \leq s \leq \min(p+1, k).
\end{aligned}$$

Remark 5.21 *For the purposes of deriving the forthcoming* a priori *error bound on the error in the computed target functional, cf. Theorem 5.23 below, it is convenient to leave the statement of the third approximation result above in terms of m_f and m_κ, rather than in terms of the stretching factors $\sigma_{i,\kappa}$, $i = 1, \ldots, d$, solely, since these quantities naturally arise within the definition of the discontinuity-penalization parameter σ defined in (126).*

In the following sections, we consider the *a priori* and *a posteriori* error analysis, respectively, of the DG finite element method (118) in terms of certain linear target functionals of practical interest.

5.6 *A priori* error bounds

In this section we derive an *a priori* error bound for the interior penalty DG method introduced in Section 5.3. Following the arguments presented in Sections 2 & 4, cf. also [77, 79], we begin our analysis by considering the following adjoint problem: find $z \in H^2(\Omega, \mathcal{T}_h)$ such that

$$\mathcal{B}(w, z) = J(w) \quad \forall w \in H^2(\Omega, \mathcal{T}_h). \tag{143}$$

Let us assume that (143) possesses a unique solution. Clearly, the validity of this assumption depends on the choice of the linear functional under consideration; see the discussion in Section 2.3 and article [77].

We now decompose the global error $u - u_h$ as

$$u - u_h = (u - \Pi_p^{\mathcal{T}_h} u) + (\Pi_p^{\mathcal{T}_h} u - u_h) \equiv \eta + \xi\,, \tag{144}$$

where $\Pi_p^{\mathcal{T}_h}$ denotes the L_2–projection operator introduced in Section 5.5. With these definitions we have the following result.

Lemma 5.22 *Assume that (113) and (129) hold and let* $\gamma_1|_\kappa = \|c/c_0\|^2_{L_\infty(\kappa)}$*; then the functions* ξ *and* η *defined by (144) satisfy the following inequality*

$$\begin{aligned} |||\xi|||^2 \;\leq\; C \Bigg(& \sum_{\kappa\in\mathcal{T}_h} \Big(\|\sqrt{a}\nabla\eta\|^2_{L_2(\kappa)} + \gamma_1\|\eta\|^2_{L_2(\kappa)} + \|\eta^+\|^2_{\partial_+\kappa\cap\Gamma} + \|\eta^-\|^2_{\partial_-\kappa\setminus\Gamma} \Big) \\ & + \int_{\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}} \frac{1}{\vartheta} |\{\!\{a\nabla\eta\}\!\}|^2 \,\mathrm{d}s + \int_{\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}} \vartheta |[\![\eta]\!]|^2 \,\mathrm{d}s \Bigg) \,, \end{aligned}$$

where C *is a positive constant that depends only on the dimension* d *and the polynomial degree* p*.*

Proof: From the Galerkin orthogonality condition (128), we deduce that

$$\mathcal{B}(\xi,\xi) = -\mathcal{B}(\eta,\xi),$$

where ξ and η are as defined in (144). Thereby, employing the coercivity result stated in Theorem 5.6 gives

$$|||\xi|||^2 \leq -\frac{1}{C}\mathcal{B}(\eta,\xi)\,. \tag{145}$$

Using the identity (130), the right–hand side of (145) may be bounded as follows:

$$\begin{aligned} \mathcal{B}(\eta,\xi) \;\leq\; C|||\xi||| \Bigg(& \sum_{\kappa\in\mathcal{T}_h} \Big(\|\sqrt{a}\nabla\eta\|^2_{L_2(\kappa)} + \gamma_1\|\eta\|^2_{L_2(\kappa)} + \|\eta^+\|^2_{\partial_+\kappa\cap\Gamma} + \|\eta^-\|^2_{\partial_-\kappa\setminus\Gamma} \Big) \\ & + \int_{\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}} \frac{1}{\vartheta} |\{\!\{a\nabla\eta\}\!\}|^2 \,\mathrm{d}s + \int_{\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}} \vartheta |[\![\eta]\!]|^2 \,\mathrm{d}s \Bigg)^{1/2} ; \end{aligned} \tag{146}$$

see [104, 74] for details. Substituting (146) into (145) gives the desired result. □

For the rest of this section, let us now assume that the volume of the elements, denoted by m_κ for each $\kappa \in \mathcal{T}_h$, has *bounded local variation*; i.e., there exists a constant $C_8 \geq 1$ such that, for any pair of elements κ and κ' which share a $(d-1)$–dimensional face,

$$C_8^{-1} \leq m_\kappa/m_{\kappa'} \leq C_8\,. \tag{147}$$

With this hypothesis, we now proceed to prove the main result of this section.

Theorem 5.23 *Let* $\Omega \subset \mathbb{R}^d$ *be a bounded polyhedral domain and* $\mathcal{T}_h = \{\kappa\}$ *a subdivision of* Ω*, such that the elemental volumes satisfy the bounded local variation condition (147). Then, assuming that conditions (113), (117), and (129) on the data hold, and* $u \in H^k(\Omega,\mathcal{T}_h)$*,* $k \geq 2$*,* $z \in H^l(\Omega,\mathcal{T}_h)$*,* $l \geq 2$*, then the solution* $u_h \in V_{h,p}$ *of (118) obeys the error bound*

$$\begin{aligned} |J(u) - J(u_h)|^2 \;\leq\; C & \left(\sum_{\kappa\in\mathcal{T}_h} \left\{ \frac{\alpha}{\sigma^2_{d,\kappa}} + \frac{\beta_2}{\sigma_{d,\kappa}} + (\beta_1+\gamma_1) \right\} \int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{u},\Sigma_\kappa,U_\kappa)\,\mathrm{d}\tilde{\mathbf{x}} \right) \\ & \times \left(\sum_{\kappa\in\mathcal{T}_h} \left\{ \frac{\alpha}{\sigma^2_{d,\kappa}} + \frac{\beta_2}{\sigma_{d,\kappa}} + (\beta_1+\gamma_2) \right\} \int_{\tilde{\kappa}} D^t_{\tilde{\kappa}}(\tilde{z},\Sigma_\kappa,U_\kappa)\,\mathrm{d}\tilde{\mathbf{x}} \right), \end{aligned}$$

for $2 \leq s \leq \min(p+1,k)$ *and* $2 \leq t \leq \min(p+1,l)$*, where* $\alpha|_\kappa = \bar{a}_{\tilde{\kappa}}$*,* $\beta_1|_\kappa = \|c+\nabla\cdot\mathbf{b}\|_{L_\infty(\kappa)}$*,* $\beta_2|_\kappa = \|\mathbf{b}\|_{L_\infty(\kappa)}$*,* $\gamma_1|_\kappa = \|c/c_0\|^2_{L_\infty(\kappa)}$*,* $\gamma_2|_\kappa = \|(c+\nabla\cdot\mathbf{b})/c_0\|^2_{L_\infty(\kappa)}$*, for all* $\kappa \in \mathcal{T}_h$*. Here,* C *is a constant depending on the dimension* d*, the polynomial degree* p*, and the parameters* C_i*,* $i = 1,\ldots,8$*.*

Proof: Decomposing the error $u - u_h$ as in (144), we note that the error in the target functional $J(\cdot)$ may be expressed as follows:

$$\begin{aligned} J(u) - J(u_h) &= J(u-u_h) \\ &= \mathcal{B}(u-u_h, z) \\ &= \mathcal{B}(u-u_h, z-z_h) \\ &= \mathcal{B}(\eta, z-z_h) + \mathcal{B}(\xi, z-z_h) \\ &\equiv \mathrm{I} + \mathrm{II} \end{aligned} \tag{148}$$

for all $z_h \in V_{h,p}$. Let us first deal with term I. To this end, we define $z_h = \Pi_p^{\mathcal{T}_h} z$ and $w = z - z_h$; after a lengthy, but straightforward calculation, we deduce that

$$\begin{aligned} \mathrm{I}^2 \leq\ & C \left(\sum_{\kappa\in\mathcal{T}_h} \Big\{ \|\sqrt{a}\nabla\eta\|^2_{L_2(\kappa)} + \beta_1\|\eta\|^2_{L_2(\kappa)} + \beta_2\epsilon_\kappa^{-1}\|\nabla\eta\|^2_{L_2(\kappa)} + \|[\![\eta]\!]\|^2_{\partial_-\kappa} \right. \\ & \left. + \|\vartheta^{-1/2}\{\!\{a\nabla\eta\}\!\}\|^2_{L_2(\partial\kappa\cap(\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}))} + \|\vartheta^{1/2}[\![\eta]\!]\|^2_{L_2(\partial\kappa\cap(\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}))} \Big\} \right) \\ & \times \left(\sum_{\kappa\in\mathcal{T}_h} \Big\{ \|\sqrt{a}\nabla w\|^2_{L_2(\kappa)} + \beta_1\|w\|^2_{L_2(\kappa)} + \beta_2\epsilon_\kappa\|w\|^2_{L_2(\kappa)} + \|w^+\|^2_{\partial_-\kappa} \right. \\ & \left. + \|\vartheta^{-1/2}\{\!\{a\nabla w\}\!\}\|^2_{L_2(\partial\kappa\cap(\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}))} + \|\vartheta^{1/2}[\![w]\!]\|^2_{L_2(\partial\kappa\cap(\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}))} \Big\} \right), \end{aligned} \tag{149}$$

for any set of real positive numbers ϵ_κ, $\kappa \in \mathcal{T}_h$. Let us now consider Term II. Here, we note that a bound analogous to (146) in the proof of Lemma 5.22 holds with η and ξ replaced by ξ and w in (146), respectively. Indeed, in this case we have that

$$\begin{aligned} |\mathcal{B}(\xi,w)| \leq\ & |||\xi||| \times \left[\sum_{\kappa\in\mathcal{T}_h} \Big(\|\sqrt{a}\nabla w\|^2_{L_2(\kappa)} + \gamma_2\|w\|^2_{L_2(\kappa)} + \|w^+\|^2_{\partial_-\kappa} \right. \\ & \left. + \|\vartheta^{1/2}[\![w]\!]\|^2_{L_2(\partial\kappa\cap(\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}))} + \|\vartheta^{-1/2}\{\!\{a\nabla w\}\!\}\|^2_{L_2(\partial\kappa\cap(\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}))} \Big) \right]^{\frac{1}{2}}. \end{aligned} \tag{150}$$

Thereby, employing Lemma 5.22 in (150) and inserting the result and (149) into (148) we

deduce that

$$
\begin{aligned}
|J(u)-J(u_h)|^2 \;\leq\; C \Bigg(& \sum_{\kappa\in\mathcal{T}_h} \Big\{ \|\sqrt{a}\nabla\eta\|^2_{L_2(\kappa)} + (\beta_1+\gamma_1)\,\|\eta\|^2_{L_2(\kappa)} + \beta_2\epsilon_\kappa^{-1}\|\nabla\eta\|^2_{L_2(\kappa)} \\
& + \|\eta^+\|^2_{\partial_+\kappa\cap\Gamma} + \|\eta^-\|^2_{\partial_-\kappa\setminus\Gamma} + \|[\![\eta]\!]\|^2_{\partial_-\kappa} \\
& + \|\vartheta^{-1/2}\{\!\{a\nabla\eta\}\!\}\|^2_{L_2(\partial\kappa\cap(\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}))} + \|\vartheta^{1/2}[\![\eta]\!]\|^2_{L_2(\partial\kappa\cap(\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}))} \Big\}\Bigg) \\
& \times \Bigg(\sum_{\kappa\in\mathcal{T}_h} \Big\{ \|\sqrt{a}\nabla w\|^2_{L_2(\kappa)} + (\beta_1+\beta_2\epsilon_\kappa+\gamma_2)\,\|w\|^2_{L_2(\kappa)} \\
& + \|w^+\|^2_{\partial_-\kappa} + \|\vartheta^{-1/2}\{\!\{a\nabla w\}\!\}\|^2_{L_2(\partial\kappa\cap(\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}))} \\
& + \|\vartheta^{1/2}[\![w]\!]\|^2_{L_2(\partial\kappa\cap(\Gamma_{\mathcal{I}}\cup\Gamma_{\mathrm{D}}))} \Big\}\Bigg)\,. \qquad (151)
\end{aligned}
$$

After application of Theorem 5.20 gives

$$
\begin{aligned}
|J(u)-J(u_h)|^2 \;\leq\; C \Bigg(& \sum_{\kappa\in\mathcal{T}_h} \Bigg\{ \frac{\bar{a}_\kappa}{\sigma^2_{d,\kappa}} \left[1 + \frac{\bar{a}_\kappa}{m_\kappa} \sum_{f\subset\partial\kappa} \frac{m_f}{\vartheta_f} + \frac{\sigma_{d,\kappa}\sum_{f\subset\partial\kappa}\vartheta_f}{\bar{a}_\kappa} \right] \\
& + \frac{\beta_2}{\sigma_{d,\kappa}} \left[1 + \frac{1}{\epsilon_\kappa\sigma_{d,\kappa}} \right] + (\beta_1+\gamma_1) \Bigg\} \int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{u},\Sigma_\kappa,U_\kappa)\,\mathrm{d}\tilde{\mathbf{x}} \Bigg) \\
& \times \Bigg(\sum_{\kappa\in\mathcal{T}_h} \Bigg\{ \frac{\bar{a}_\kappa}{\sigma^2_{d,\kappa}} \left[1 + \frac{\bar{a}_\kappa}{m_\kappa} \sum_{f\subset\partial\kappa} \frac{m_f}{\vartheta_f} + \frac{\sigma_{d,\kappa}\sum_{f\subset\partial\kappa}\vartheta_f}{\bar{a}_\kappa} \right] \\
& + \frac{\beta_2}{\sigma_{d,\kappa}} \left[1 + \epsilon_\kappa\sigma_{d,\kappa} \right] + (\beta_1+\gamma_2) \Bigg\} \int_{\tilde{\kappa}} D^t_{\tilde{\kappa}}(\tilde{z},\Sigma_\kappa,U_\kappa)\,\mathrm{d}\tilde{\mathbf{x}} \Bigg)\,.
\end{aligned}
$$

The statement of Theorem 5.23 now follows by selecting $\epsilon_\kappa = 1/\sigma_{d,\kappa}$, for each $\kappa\in\mathcal{T}_h$, and employing the definition of the discontinuity-penalization parameter ϑ stated in (126), together with the bounded variation of the elemental volumes (147) and (140). □

Remark 5.24 *The above result represents an extension of the* a priori *error bound derived in the article [50] to the case when general anisotropic computational meshes are employed. We note that although the analysis presented in [50] assumed shape–regular meshes, the explicit dependence of the polynomial degree was retained in the resulting* a priori *error bound; however, following the arguments in [50] an analogous hp–version bound of the form stated in Theorem 5.23 may be deduced; this will be considered in detail in Section 6.*

Remark 5.25 *The* a priori *bound stated in Theorem 5.23 clearly highlights that in order to minimize the error in the computed target functional $J(\cdot)$, the design of an optimal mesh must exploit anisotropic information emanating from both the primal and adjoint solutions u and z, respectively. Indeed, a mesh solely optimized for u may be completely inappropriate for z, and vice verse, thus there must be a trade-off between aligning the elements with respect to either solution in order to minimize the overall error in $J(\cdot)$.*

5.7 *A posteriori* error estimation and adaptivity

In this section we consider the derivation of an adjoint-based *a posteriori* error bound for the error in the computed target functional $J(u_h)$, together with its implementation into a general adaptive algorithm in the anisotropic setting.

For a given linear functional $J(\cdot)$ the proceeding *a posteriori* error bound will be expressed in terms of the finite element residual $R_{\rm int}$ defined on $\kappa \in \mathcal{T}_h$ by

$$R_{\rm int}|_\kappa = (f - Lu_h)|_\kappa \,,$$

which measures the extent to which u_h fails to satisfy the differential equation on the union of the elements κ in the mesh $\mathcal{T}_h$; thus we refer to $R_{\rm int}$ as the *internal residual*. Also, since u_h only satisfies the boundary conditions approximately, the differences $g_{\rm D} - u_h$ and $g_{\rm N} - (a\nabla u_h)\cdot\mathbf{n}$ are not necessarily zero on $\Gamma_{\rm D}\cup\Gamma_-$ and $\Gamma_{\rm N}$, respectively; thus we define the *boundary residuals* $R_{\rm D}$ and $R_{\rm N}$, respectively, by

$$R_{\rm D}|_{\partial\kappa\cap(\Gamma_{\rm D}\cup\Gamma_-)} = (g_{\rm D} - u_h^+)|_{\partial\kappa\cap(\Gamma_{\rm D}\cup\Gamma_-)}\,, \quad R_{\rm N}|_{\partial\kappa\cap\Gamma_{\rm N}} = (g_{\rm N} - (a\nabla u_h^+)\cdot\mathbf{n})|_{\partial\kappa\cap\Gamma_{\rm N}}.$$

With this notation, we may derive the following general result.

Theorem 5.26 *Let u and u_h denote the solutions of (110), (112) and (118), respectively, and suppose that the adjoint solution z is defined by (143). Then, the following error representation formula holds:*

$$J(u) - J(u_h) = \mathcal{R}(u_h, z - z_h) \equiv \sum_{\kappa\in\mathcal{T}_h} \eta_\kappa \,, \tag{152}$$

where

$$\begin{aligned}
\eta_\kappa \;=\; & \int_\kappa R_{\rm int}(z - z_h)\,{\rm d}\mathbf{x} - \int_{\partial_-\kappa\cap\Gamma} (\mathbf{b}\cdot\mathbf{n}_\kappa)\, R_{\rm D}(z - z_h)^+\,{\rm d}s \\
& + \int_{\partial_-\kappa\setminus\Gamma} \mathbf{b}\cdot[\![u_h]\!](z - z_h)^+\,{\rm d}s - \int_{\partial\kappa\cap\Gamma_{\rm D}} R_{\rm D}((a\nabla(z - z_h)^+)\cdot\mathbf{n}_\kappa)\,{\rm d}s \\
& + \int_{\partial\kappa\cap\Gamma_{\rm D}} \vartheta R_{\rm D}(z - z_h)^+\,{\rm d}s + \int_{\partial\kappa\cap\Gamma_{\rm N}} R_{\rm N}(z - z_h)^+\,{\rm d}s - \int_{\partial\kappa\setminus\Gamma} \vartheta[\![u_h]\!]\cdot\mathbf{n}_\kappa (z - z_h)^+\,{\rm d}s \\
& + \frac{1}{2}\int_{\partial\kappa\setminus\Gamma} \left\{\, [\![u_h]\!]\cdot(a\nabla(z - z_h)^+) - [\![a\nabla u_h]\!](z - z_h)^+ \right\}\,{\rm d}s
\end{aligned} \tag{153}$$

for all $z_h \in V_{h,p}$.

Proof: The proof follows from the arguments presented in Section 4.1. □

Thereby, on application of the triangle inequality, we deduce the following *a posteriori* error bound.

Corollary 5.27 *Under the assumptions of Theorem 5.26, the following* a posteriori *error bound holds:*

$$|J(u) - J(u_h)| \le \mathcal{R}_{|\Omega|}(u_h, z - z_h) \equiv \sum_{\kappa\in\mathcal{T}_h} |\eta_\kappa| \,, \tag{154}$$

where η_κ is defined as in (153).

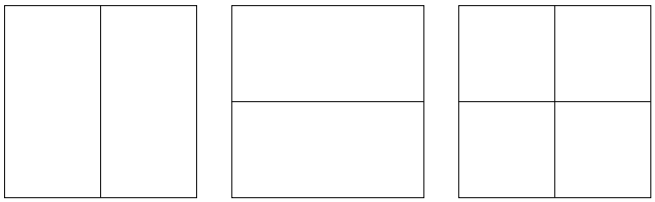


Figure 30: Cartesian refinement in 2D: (a) & (b) Anisotropic refinement; (c) Isotropic refinement.

For a user-defined tolerance TOL, we now consider the problem of designing an appropriate finite element mesh $\mathcal{T}_h$ such that

$$|J(u) - J(u_h)| \leq \texttt{TOL}\,,$$

subject to the constraint that the total number of elements in $\mathcal{T}_h$ is minimized. For simplicity of presentation, in this section we first consider the case when $\Omega \subset \mathbb{R}^2$ and $\mathcal{T}_h$ consists of *1–irregular* quadrilateral elements; the generalization to hexahedral meshes when $\Omega \subset \mathbb{R}^3$ will be treated at the end of this section. Following the discussion presented in [77], we exploit the *a posteriori* error bound (154) with z replaced by a suitable numerical approximation, denoted by $\bar{z}_h$, cf. Section 4.1. On the basis of numerical experimentation, we compute $\bar{z}_h \in V_{h,\bar{p}}$, $\bar{p} = p + p_{\rm inc}$; in Section 5.8, we set $p_{\rm inc} = 1$, cf. [58, 79]. Thereby, in practice we enforce the stopping criterion

$$\mathcal{R}_{|\Omega|}(u_h, \bar{z}_h - z_h) \leq \texttt{TOL}\,. \tag{155}$$

If (155) is not satisfied, then the elements are marked for refinement/derefinement according to the size of the (approximate) error indicators $|\bar{\eta}_\kappa|$; these are defined analogously to $|\eta_\kappa|$ in (153) with z replaced by $\bar{z}_h$, cf. Algorithm 4.1.

To subdivide the elements which have been flagged for refinement, we employ a simple Cartesian refinement strategy; here, elements may be subdivided either anisotropically or isotropically according to the three refinements (in two–dimensions, i.e., $d = 2$) depicted in Figure 30. In order to determine the optimal refinement, stimulated by the articles [96, 98], we propose the following strategy based on choosing the most competitive subdivision of κ from a series of trial refinements, whereby an approximate local error indicator on each trial patch is determined.

Algorithm 5.1 *Given an element κ in the computational mesh $\mathcal{T}_h$ (which has been marked for refinement), we first construct the mesh patches $\mathcal{T}_{h,i}$, $i = 1, 2, 3$, based on refining κ according to Figures 30(a), (b), & (c), respectively. On each mesh patch, $\mathcal{T}_{h,i}$, $i = 1, 2, 3$, we compute the approximate error estimators*

$$\mathcal{R}_{\kappa,i}(u_{h,i}, \bar{z}_{h,i} - z_h) = \sum_{\kappa' \in \mathcal{T}_{h,i}} \eta_{\kappa',i},$$

for $i = 1, 2, 3$, respectively. Here, $u_{h,i}$, $i = 1, 2, 3$, is the DG approximation to (110), (112) computed on the mesh patch $\mathcal{T}_{h,i}$, $i = 1, 2, 3$, respectively, based on enforcing appropriate

boundary conditions on $\partial\kappa$ computed from the original DG solution u_h on the portion of the boundary $\partial\kappa$ of κ which is interior to the computational domain Ω, i.e., where $\partial\kappa \cap \Gamma = \emptyset$. Similarly, $\bar{z}_{h,i}$ denotes the DG approximation to z computed on the local mesh patch $\mathcal{T}_{h,i}$, $i = 1, 2, 3$, respectively, with polynomials of degree $\bar{p}$, based on employing suitable boundary conditions on $\partial\kappa \cap \Gamma = \emptyset$ derived from $\bar{z}_h$. Finally, $\eta_{\kappa',i}$, $i = 1, 2, 3$, is defined in an analogous manner to η_κ, cf. (153) above, with u_h and z replaced by $u_{h,i}$ and $\bar{z}_{h,i}$, respectively.

The element κ is then refined according to the subdivision of κ which satisfies

$$\min_{i=1,2,3} \frac{|\eta_\kappa| - |\mathcal{R}_{\kappa,i}(u_{h,i}, \bar{z}_{h,i} - z_h)|}{\#dofs(\mathcal{T}_{h,i}) - \#dofs(\kappa)},$$

where $\#dofs(\kappa)$ and $\#dofs(\mathcal{T}_{h,i})$, $i = 1, 2, 3$, denote the number of degrees of freedom associated with κ and $\mathcal{T}_{h,i}$, $i = 1, 2, 3$, respectively.

An alternative approach which is very similar to Algorithm 5.1 is to simply construct the mesh patches $\mathcal{T}_{h,i}$, $i = 1, 2$, and compute the approximate local primal and adjoint solutions on these meshes only. Given an anisotropy parameter $\theta \geq 1$, isotropic refinement is selected when

$$\frac{\max_{i=1,2} |\mathcal{R}_{\kappa,i}(u_{h,i}, \bar{z}_{h,i} - z_h)|}{\min_{i=1,2} |\mathcal{R}_{\kappa,i}(u_{h,i}, \bar{z}_{h,i} - z_h)|} < \theta;$$

otherwise an anisotropic refinement is performed based on which refinement gives rise to the smallest predicted error indicator, i.e., the subdivision for which $|\mathcal{R}_{\kappa,i}(u_{h,i}, \bar{z}_{h,i} - z_h)|$, $i = 1, 2$, is minimal. For the purposes of these lecture notes, we shall not pursue this latter approach; rather, we refer the reader to [43, 49] for details.

The extension of this approach to the case when $\mathcal{T}_h$ is a hexahedral mesh in three-dimensions follows in an analogous fashion. Indeed, in this setting, we again employ a Cartesian refinement strategy whereby elements may be subdivided either isotropically or anisotropically according to the four refinements depicted in Figures 31(a)–(d). We remark that we assume that a face in the computational mesh is a complete face of at least one element. This assumption means that the refinements depicted in Figures 30(b)–(d) may be inadmissible. In this situation, we replace the selected refinement by either one of the anisotropic mesh refinements depicted in Figures 31(e)–(g), or if necessary, an isotropic refinement is performed.

5.8 Numerical experiments

In this section we present a number of experiments to numerically demonstrate the performance of the anisotropic adaptive algorithm outlined in Section 5.7.

5.8.1 Singularly perturbed advection-diffusion problem

In this first example we consider a linear singularly perturbed advection-diffusion problem on the (unit) square domain $\Omega = (0, 1)^2$, where $a = \varepsilon I$, $0 < \varepsilon \ll 1$, $\mathbf{b} = (1, 1)^\top$, $c = 0$, and f is chosen so that

$$u(x, y) = x + y(1 - x) + [\mathrm{e}^{-1/\varepsilon} - \mathrm{e}^{-(1-x)(1-y)/\varepsilon}]\,[1 - \mathrm{e}^{-1/\varepsilon}]^{-1}, \tag{156}$$

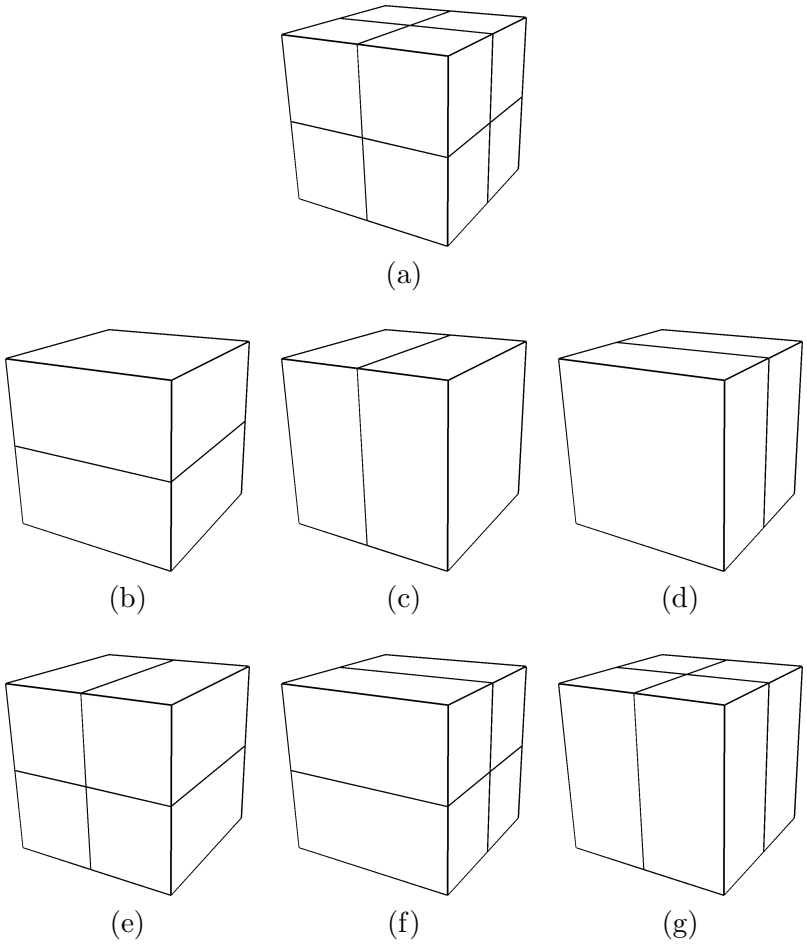
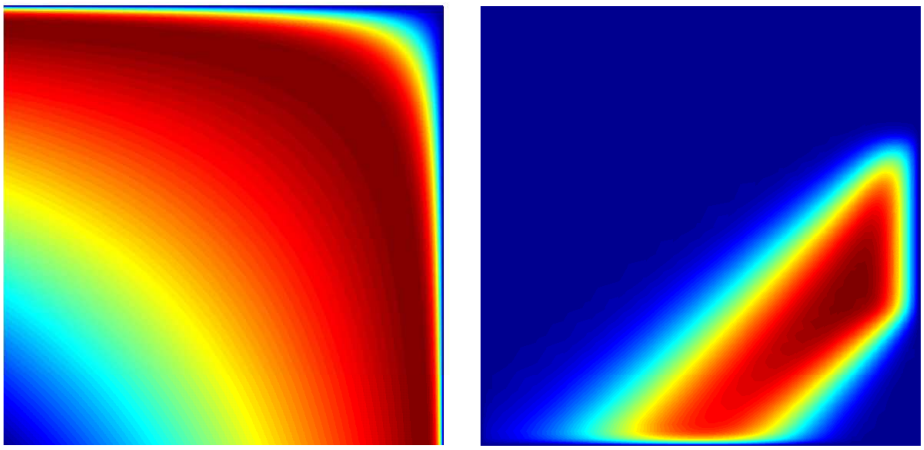


Figure 31: Cartesian refinement in 3D.

cf. [74]. For $0 < \varepsilon \ll 1$, solution (156) has boundary layers along $x = 1$ and $y = 1$. Throughout this section we set $\varepsilon = 10^{-2}$; Figure 32(a) shows the analytical solution in this case.

Here, we suppose that the aim of the computation is to calculate the (weighted) mean value of u over Ω, i.e., $J(u) = \int_\Omega u\psi \,\mathrm{d}\mathbf{x}$, where $\psi = 100(1 - \tanh(100(r_1 - 0.01)(r_1 + 0.01)))(1 - \tanh(100(r_2 - 0.2)(r_2 + 0.2)))$, $r_1 = x - 1.0$ and $r_2 = y - 0.5$; thereby, $J(u) = 4.409917162888037$. The corresponding analytical solution to the adjoint problem is depicted in Figure 32(b).

To demonstrate the versatility of the proposed competitive refinement algorithm, cf. Algorithm 5.1, in this section we employ bi-linear, bi-quadratic, and bi-cubic elements, i.e., $p = 1$, $p = 2$, and $p = 3$, respectively. To this end, in Figure 33 we plot the error in the computed target functional $J(\cdot)$ using both an isotropic (only) mesh refinement algorithm, together with the anisotropic refinement strategy outlined in Section 5.7. For purposes of comparison with standard anisotropic refinement strategies employed within the literature, we also consider the use of a Hessian–based algorithm. More precisely, for each element in the mesh, we construct a metric for the primal and adjoint problems based on computing the pos-



(a) (b)

Figure 32: Advection–diffusion problem with $\epsilon = 10^{-2}$: (a) Primal solution; (b) Adjoint solution.

itive part of the Hessian matrix of the computed numerical solutions u_h and $\bar{z}_h$, respectively. Upon application of the metric intersection algorithm proposed in [28], elements marked for refinement are anisotropically/isotropically subdivided, as in Figure 30, according to the relative size of the eigenvalues of the newly constructed metric; see [40] for details. We point out that this general strategy is based on minimizing anisotropic *a priori* bounds on the error between the unknown analytical solution and its numerical approximation computed using piecewise linear polynomials, *assuming* that the analytical solution is sufficiently regular. In practice, this type of anisotropic mesh refinement strategy, based on computing second-order derivatives of the numerical solution, has proven to be extremely successful, though its extension to higher-order polynomials still remains an open question. The purpose of the following section is to demonstrate that the newly proposed anisotropic refinement algorithm is both competitive with the classical Hessian approach when piecewise linear elements are employed, but also that they lead to the design of computationally efficient meshes when higher-order polynomial degrees are exploited.

Firstly, for each polynomial degree employed, we clearly observe the superiority of employing the competitive anisotropic mesh refinement algorithm in comparison with standard isotropic subdivision of the elements. Indeed, the error $|J(u) - J(u_h)|$ computed on the series of anisotropically refined meshes designed using Algorithm 5.1 outlined in Section 5.7 is always less than the corresponding quantity computed on the isotropic grids. Here, we observe that there is an initial transient whereby the error in the computed target functional decays rapidly using the former refinement algorithm, in comparison with the latter, after which the gradient of the convergence curves become very similar. This type of behavior is indeed expected, since for a fixed order method, i.e., h–version, we can only expect to improve the convergence of the error by a fixed constant, as the mesh is refined. Notwithstanding this, we note that, for each polynomial degree employed, the true error between $J(u)$ and $J(u_h)$ using anisotropic refinement is around an order of magnitude smaller than the corresponding quantity when isotropic refinement is employed alone. Secondly, we observe that for all

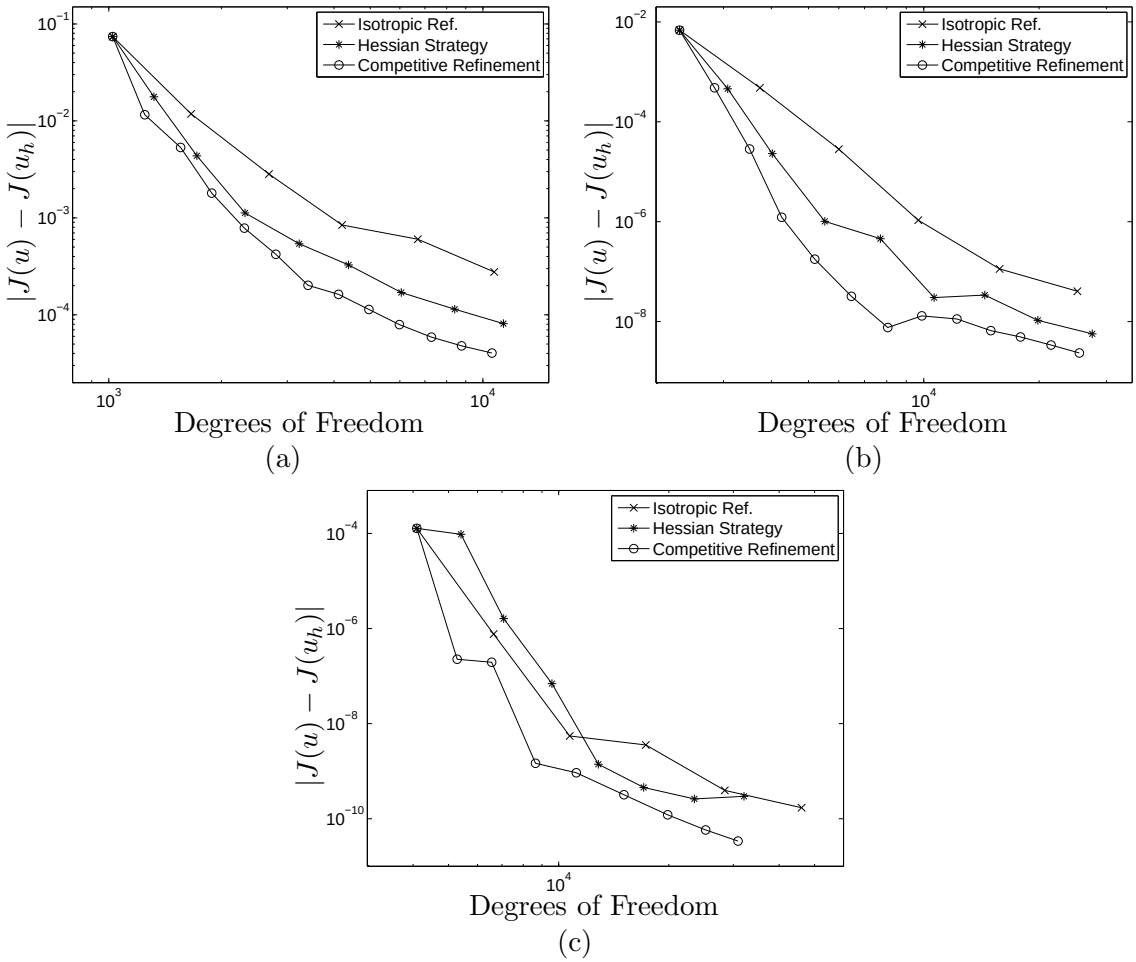
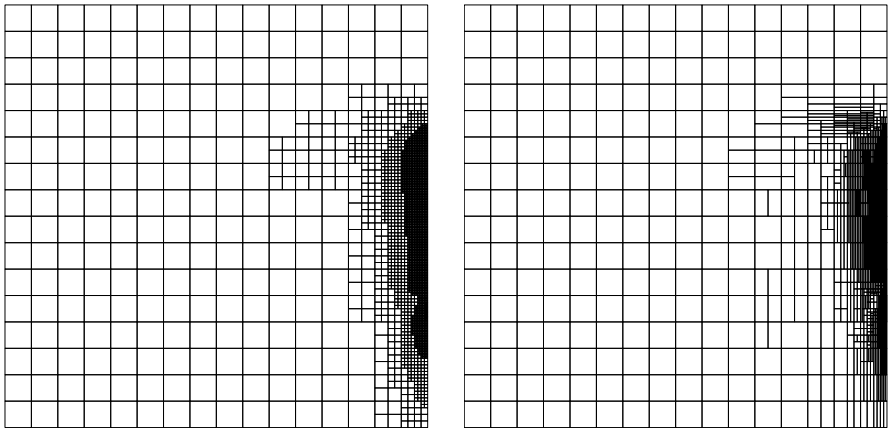


Figure 33: Advection–diffusion problem with $\epsilon = 10^{-2}$: Comparison between adaptive isotropic and anisotropic mesh refinement. (a) $p = 1$; (b) $p = 2$; (c) $p = 3$.

polynomial degrees employed, the Hessian strategy is inferior to Algorithm 5.1, in the sense that the error in the target functional computed using the latter strategy is always smaller than the corresponding quantity computed using the former strategy, for a fixed number of degrees of freedom. Indeed, even for bi-linear elements, for which the Hessian strategy has been proposed on the basis of interpolation theory, Algorithm 5.1 leads to a 35% reduction in the error on the final mesh in comparison with the corresponding quantity computed using the Hessian-based approach. Similar behavior is also observed for bi-quadratic and bi-cubic elements, though in the latter case, the Hessian strategy actually generates meshes which in many cases are inferior to their isotropic counterparts.

In Figure 34 we show the meshes generated using both isotropic and anisotropic mesh adaptation. For brevity, we only show the meshes for $p = 1$, and in the latter case employing Algorithm 5.1. Firstly, we note that in both cases the mesh is primarily concentrated in the vicinity of the boundary layer along $x = 1$, where the support of the weighting function ψ appearing in the definition of the target functional $J(\cdot)$ is non-zero. Indeed, the region of the



(a) (b)

Figure 34: Advection–diffusion problem with $\epsilon = 10^{-2}$: Adaptively refined meshes for $p = 1$. (a) Isotropic mesh after 5 adaptive refinements, with 2680 elements; (b) Anisotropic mesh designed using Algorithm 5.1 after 7 adaptive refinements, with 963 elements

computational domain where the remainder of the boundary layer along $x = 1$ and moreover where the boundary layer along $y = 1$ are located are not refined, since the resolution of these sharp features present in the analytical solution are not important for the accurate computation of the selected target functional, cf. [58], for example. For Algorithm 5.1, we observe that the refinement strategy has clearly identified the anisotropy in the underlying primal and adjoint solutions, and refined the mesh accordingly. Indeed, we observe that the boundary layer along $x = 1$, $0 \leq y \leq 1$, has been significantly refined, as we would expect, with the elements being mostly refined in the direction parallel to the boundary. We note, however, that some anisotropic refinement perpendicular to Γ is performed in the region of the boundary layer in order to accurately capture the anisotropy of the adjoint solution z.

5.8.2 ADIGMA MTC3: Laminar flow around a NACA0012 airfoil

In this example, we consider the subsonic viscous flow around a NACA0012 airfoil; here, the upper and lower surfaces of the airfoil geometry are specified by the function $g^{\pm}$, respectively, where

$$g^{\pm}(s) = \pm 5 \times 0.12 \times (0.2969 s^{1/2} - 0.126 s - 0.3516 s^2 + 0.2843 s^3 - 0.1015 s^4).$$

As the chord length l of the airfoil is $l \approx 1.00893$ we use a rescaling of g in order to yield an airfoil of unit (chord) length. At the farfield (inflow) boundary we specify a Mach 0.5 flow at an angle of attack $\alpha = 2^{\circ}$, with Reynolds number $\mathrm{Re} = 5000$; on the walls of the airfoil geometry, we impose a zero heat flux (adiabatic) no-slip boundary condition. This is a standard laminar test case which has been investigated by many other authors, cf. [14, 61], for example, and serves as one of the test cases for the EU project ADIGMA [82].

Here, we consider the estimation of the drag coefficient C_{d}; i.e., the target functional of interest is given by

$$J(\cdot) \equiv J_{C_{\mathrm{d}}}(\cdot),$$

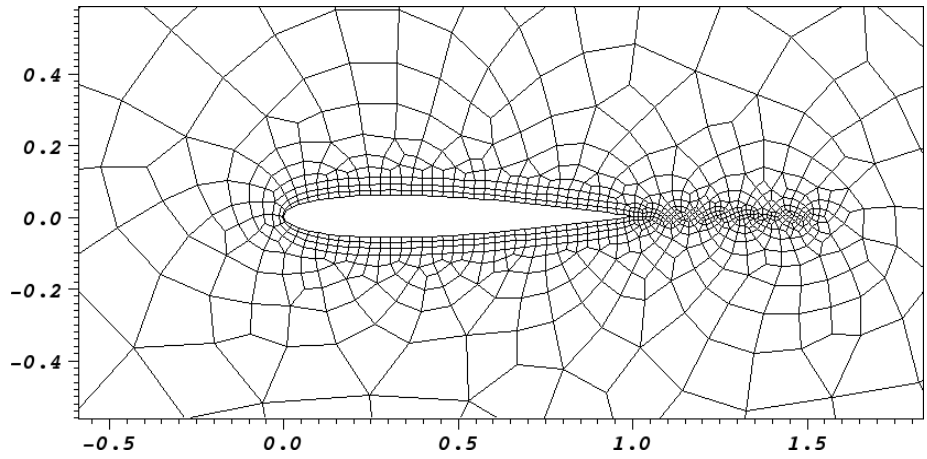


Figure 35: ADIGMA MTC3 test case: Zoom of initial mesh with 1134 elements.

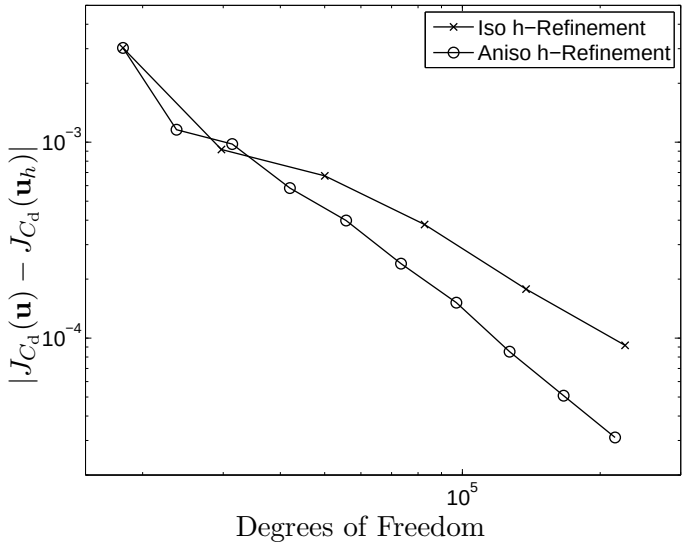
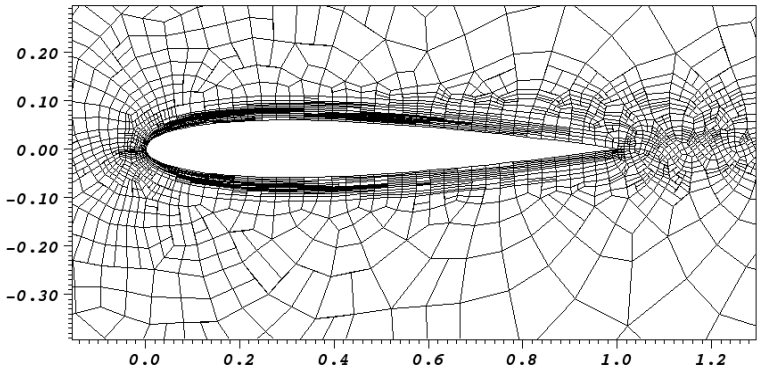
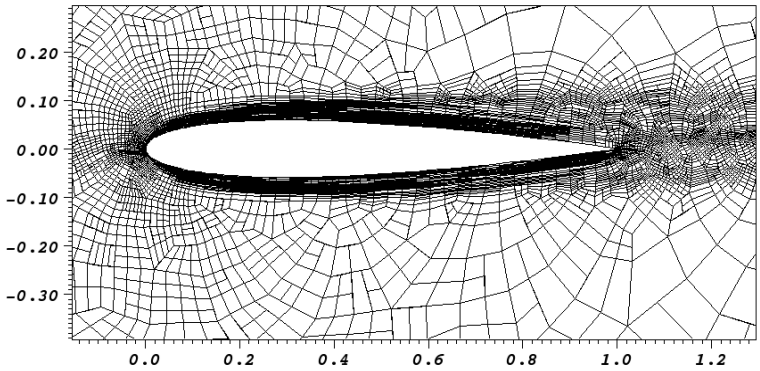


Figure 36: ADIGMA MTC3 test case: Comparison between adaptive isotropic and anisotropic mesh refinement.



(a)



(b)

Figure 37: ADIGMA MTC3 test case: Anisotropic mesh after (a) 4 adaptive refinements, with 3485 elements; (b) 8 adaptive refinements, with 10410 elements.

where $J_{C_\mathrm{d}}(\cdot)$ is defined as the adjoint consistent approximation to C_d, cf. (66).

In this example, the initial starting mesh is taken to be an unstructured quadrilateral–dominant hybrid mesh consisting of both quadrilateral and triangular elements; here, the total number of elements is 1134, cf. Figure 35. Here, curved boundaries are approximated by piecewise quadratic polynomials. In Figure 36 we plot the error in the computed target functional $J_{C_\mathrm{d}}(\cdot)$ using both an isotropic (only) mesh refinement algorithm, together with the anisotropic refinement strategy outlined in Section 5.7. From Figure 36, we again observe the superiority of employing the anisotropic mesh refinement algorithm in comparison with standard isotropic subdivision of the elements. Indeed, the error $|J_{C_\mathrm{d}}(\mathbf{u}) - J_{C_\mathrm{d}}(\mathbf{u}_h)|$ computed on the series of anisotropically refined meshes designed using the proposed algorithm outlined in Section 5.7 is (almost) always less than the corresponding quantity computed on the isotropic grids. Indeed, on the final mesh anisotropic mesh refinement leads to an improvement in $|J_{C_\mathrm{d}}(\mathbf{u}) - J_{C_\mathrm{d}}(\mathbf{u}_h)|$ of over 60% compared with the same quantity computed using isotropic mesh refinement. The meshes generated after 4 and 8 anisotropic adaptive mesh refinements are shown in Figure 37. Here, we clearly observe significant anisotropic refinement of the viscous boundary layer, as we would expect.

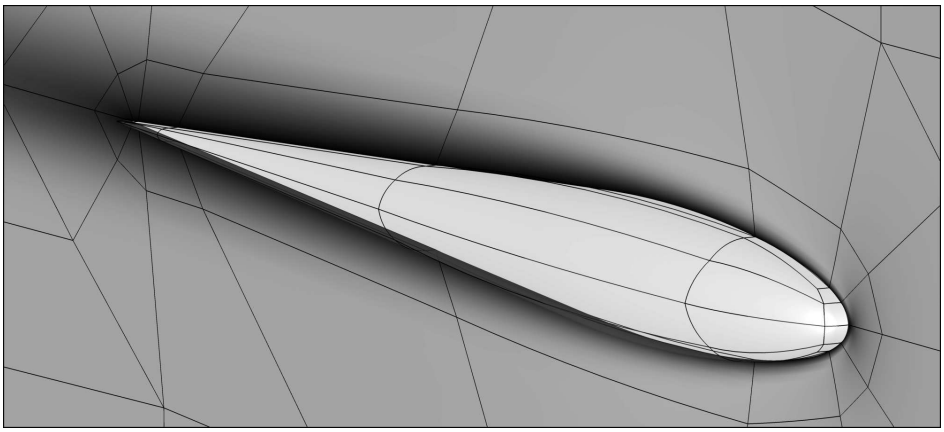


Figure 38: ADIGMA BTC0 test case at laminar conditions: Initial coarse mesh on the body surface and the symmetry plane. The symmetry plane coloring is based on the Mach number distribution computed on a fine mesh, [64].

5.8.3 ADIGMA BTC0: Laminar flow around streamlined body

In this final example we consider laminar flow past a streamlined three–dimensional body. Here, the geometry of the body is based on a 10 percent thick airfoil with boundaries constructed by a surface of revolution, see Figure 38. The BTC0 geometry is considered at laminar conditions with inflow Mach number equal to 0.5, at an angle of attack $\alpha = 1^{\circ}$, and Reynolds number $\mathrm{Re} = 5000$ with adiabatic no-slip wall boundary condition imposed. This test case has been defined in the EU project ADIGMA [82] to enable convergence studies.

Here, we suppose that the aim of the computation is to calculate the lift coefficient C_{l}; i.e., $J(\cdot) \equiv J_{C_{\mathrm{l}}}(\cdot)$, cf. (66). In this example, the initial starting mesh is taken to be an unstructured hexahedral mesh with 992 elements. In Figure 39 we plot the error in the computed target functional $J_{C_{\mathrm{l}}}(\cdot)$ using both an isotropic (only) mesh refinement algorithm, together with the anisotropic refinement strategy outlined in Section 5.7. From Figure 39, we again observe the superiority of employing the anisotropic mesh refinement algorithm in comparison with standard isotropic subdivision of the elements. Indeed, the error $|J_{C_{\mathrm{l}}}(\mathbf{u}) - J_{C_{\mathrm{l}}}(\mathbf{u}_h)|$ computed on the series of anisotropically refined meshes designed using Algorithm 5.1 is always less than the corresponding quantity computed on the isotropic grids. Indeed, on the final mesh the true error between $J_{C_{\mathrm{l}}}(\mathbf{u})$ and $J_{C_{\mathrm{l}}}(\mathbf{u}_h)$ using anisotropic mesh refinement is over an order of magnitude smaller than the corresponding quantity when isotropic h–refinement is employed alone. The mesh generated after 3 anisotropic adaptive mesh refinements is shown in Figure 40. Here, we clearly observe significant anisotropic refinement of the viscous boundary layer, as we would expect.

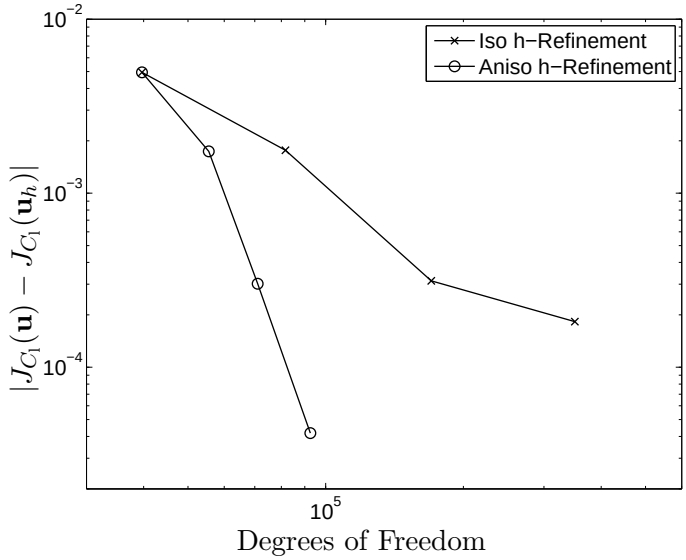
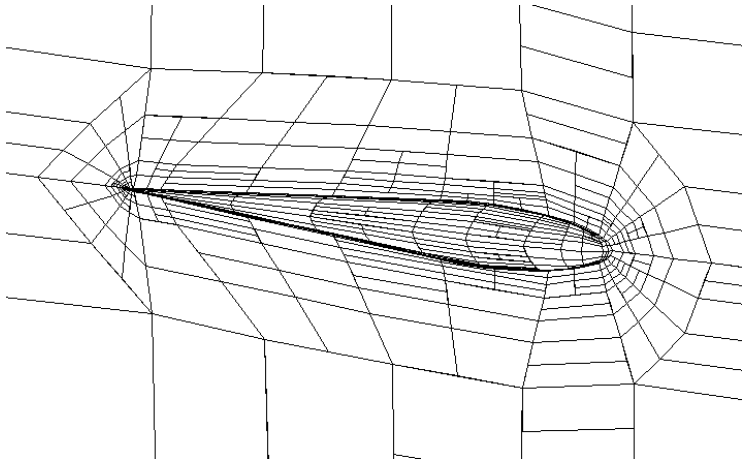


Figure 39: ADIGMA BTC0 test case (laminar): Comparison between adaptive isotropic and anisotropic mesh refinement.

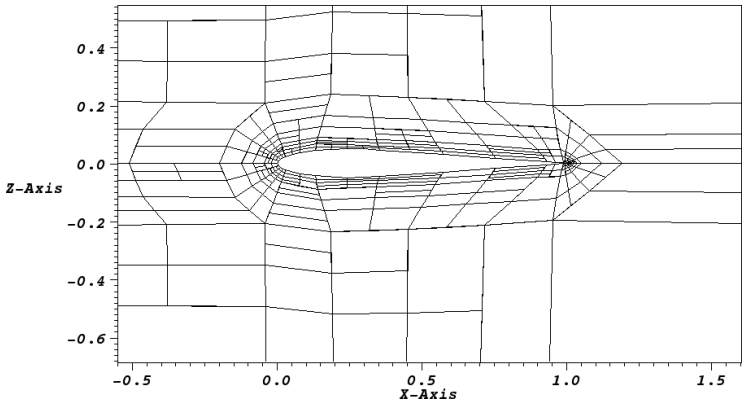
6 High-order/hp–adaptive finite element methods for compressible flows

Adaptive finite element methods that exploit both local polynomial–degree variation (p–refinement) and local mesh subdivision (h–refinement) offer much greater flexibility and improved efficiency than mesh refinement algorithms which only incorporate h–refinement or p–refinement in isolation. Indeed, since the early analytical paper of Gui and Babuška [48], the benefits of hp–version finite element methods have been clearly established for elliptic boundary value problems (see, for example, the monograph of Schwab [99]), particularly in the field of linear elasticity. The application of hp–version finite element methods to hyperbolic/nearly–hyperbolic problems is less standard, although their potential in compressible gas dynamics was first demonstrated by J.E. Flaherty and collaborators (see [26, 33], for example); for more recent work in this area, we refer to our series of papers [72, 75, 77, 102, 103], for example. The argument in favour of using an hp–version finite element method for the numerical solution of a hyperbolic/nearly–hyperbolic equation rests on the observation that while solutions to these equations may exhibit local singularities and discontinuities, in large parts of the computational domain the solution is typically a real analytic function. Such large variations in the smoothness of the solution can be captured in a particularly simple and flexible manner by using a finite element method based on discontinuous piecewise polynomials, such as the DG finite element method.

In this section we extend the error analysis developed in Section 5 for interior penalty DG methods applied to second–order partial differential equations with nonnegative characteristic form to the case when general finite element spaces are employed which allow for anisotropy in possibly both the local meshsize and the local polynomial degree. The proofs of the *a priori* error bounds presented in this section are based on exploiting the analysis developed



(a)



(b)

Figure 40: ADIGMA BTC0 test case (laminar). Anisotropic mesh after 3 adaptive refinements, with 2314 elements: (a) Boundary mesh; (b) Symmetry plane.

in Section 5, which assumed that the underlying polynomial approximation order is uniform over the computational mesh, together with the *hp*–approximation results presented in [74] and [42].

The discussion presented in this section is a brief survey of the articles [44, 45]; see also [49].

6.1 Model problem and discretization

In this section we briefly recall the model problem and interior penalty DG method introduced in Section 5.1. To this end, let Ω be a bounded open polyhedral domain in $\mathbb{R}^d$, $d = 2, 3$, and let Γ signify the union of its $(d-1)$–dimensional open edges/faces, respectively. We consider the second–order partial differential equation with nonnegative characteristic form

$$\begin{aligned} Lu \equiv -\nabla \cdot (a\nabla u) + \nabla \cdot (\mathbf{b}u) + cu &= f \,, && (157)\\ u &= g_{\mathrm{D}} \quad \text{on} \quad \Gamma_{\mathrm{D}} \cup \Gamma_{-} \,, && (158)\\ (a\nabla u) \cdot \mathbf{n} &= g_{\mathrm{N}} \quad \text{on} \quad \Gamma_{\mathrm{N}} \,, && (159) \end{aligned}$$

where $f \in L_2(\Omega)$ and $c \in L_\infty(\Omega)$ are real–valued, $\mathbf{b} = \{b_i\}_{i=1}^d$ is a vector function whose entries b_i are Lipschitz continuous real–valued functions on $\bar{\Omega}$, and $a = \{a_{ij}\}_{i,j=1}^d$ is a *symmetric* matrix whose entries a_{ij} are bounded, piecewise continuous real–valued functions defined on $\bar{\Omega}$, with

$$\boldsymbol{\zeta}^\top a(\mathbf{x})\boldsymbol{\zeta} \geq 0 \quad \forall \boldsymbol{\zeta} \in \mathbb{R}^d, \quad \text{a.e. } \mathbf{x} \in \bar{\Omega} \,. \tag{160}$$

Again, as before, we have

$$\begin{aligned} \Gamma_0 &= \{\mathbf{x} \in \Gamma : \ \mathbf{n}(\mathbf{x})^\top a(\mathbf{x})\mathbf{n}(\mathbf{x}) > 0\} \,,\\ \Gamma_- &= \{\mathbf{x} \in \Gamma \backslash \Gamma_0 : \ \mathbf{b}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0\} \,,\\ \Gamma_+ &= \{\mathbf{x} \in \Gamma \backslash \Gamma_0 : \ \mathbf{b}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) \geq 0\} \,. \end{aligned}$$

Additionally, we assume throughout that

$$(c_0(\mathbf{x}))^2 \equiv c(\mathbf{x}) + \frac{1}{2} \nabla \cdot \mathbf{b}(\mathbf{x}) \geq 0 \ \text{ a.e. } \mathbf{x} \in \Omega \,. \tag{161}$$

6.1.1 Meshes and finite element spaces

Let $\mathcal{T}_h = \{\kappa\}$ be a subdivision of the (polyhedral) domain Ω into disjoint open element domains κ constructed through the use of the mappings $Q_\kappa \circ F_\kappa$, where $F_\kappa : \hat{\kappa} \to \tilde{\kappa}$ is an affine mapping from the reference element $\hat{\kappa}$ to $\tilde{\kappa}$, and $Q_\kappa : \tilde{\kappa} \to \kappa$ is a C^1–diffeomorphism from $\tilde{\kappa}$ to the physical element κ. For simplicity of presentation, throughout this section we shall assume that $\hat{\kappa}$ is the hypercube $(-1, 1)^d$. The mapping F_κ defines the size and orientation of the element κ, while Q_κ defines the shape of κ, without any significant rescaling, or indeed change of orientation, cf. Figure 27 for the case when $d = 2$ and $\hat{\kappa} = (-1, 1)^2$. With this in mind, we assume that the element mapping Q_κ is close to the identity, cf. Section 5.2 for details.

On the reference element $\hat{\kappa}$ we define the polynomial space $\mathcal{Q}_{\vec{p}}$ with respect to the anisotropic polynomial degree vector $\vec{p} := \{p_i\}_{i=1,\ldots,d}$ as follows:

$$\mathcal{Q}_{\vec{p}} := \mathrm{span}\{\Pi_{i=1}^d \hat{x}_i^j : 0 \leq j \leq p_i\}.$$

With this notation, we introduce the following (anisotropic) finite element space.

Definition 6.1 *Let $\vec{\mathbf{p}} = (\vec{p}_\kappa : \kappa \in \mathcal{T}_h)$ be the composite polynomial degree vector of the elements in a given finite element mesh $\mathcal{T}_h$. We define the finite element space with respect to Ω, $\mathcal{T}_h$, and $\vec{\mathbf{p}}$ by*

$$V_{h,\vec{\mathbf{p}}} \quad = \quad \{u \in L_2(\Omega) : u|_\kappa \circ Q_\kappa \circ F_\kappa \in \mathcal{Q}_{\vec{p}_\kappa}\}.$$

In the special case when the elemental polynomial degree vector $\vec{p}_\kappa = \{p_{\kappa,i}\}_{i=1,\ldots,d}$, $\kappa \in \mathcal{T}_h$, is isotropic in the sense that

$$p_{\kappa,1} = p_{\kappa,2} = \ldots = p_{\kappa,d} \equiv p_\kappa$$

for all elements κ in the finite element mesh $\mathcal{T}_h$, then we write $V_{h,\mathbf{p}_{\rm iso}}$ in lieu of $V_{h,\vec{\mathbf{p}}}$, where $\mathbf{p}_{\rm iso} = (p_\kappa : \kappa \in \mathcal{T}_h)$. Clearly, in the case when the polynomial degree is both isotropic and uniformly distributed over the mesh $\mathcal{T}_h$, i.e., when $p_\kappa = p$ for all κ in $\mathcal{T}_h$, then both $V_{h,\vec{\mathbf{p}}}$ and $V_{h,\mathbf{p}_{\rm iso}}$ correspond to the finite element space $V_{h,p}$ introduced in Section 5.3.

With this notation, we now recall the DG discretization of (157)–(159): find u_h in $V_{h,\vec{\mathbf{p}}}$ such that

$$\mathcal{B}(u_h, v) = \ell(v) \tag{162}$$

for all $v \in V_{h,\vec{\mathbf{p}}}$. Here, we recall that the bilinear form $\mathcal{B}(\cdot,\cdot)$ is defined by

$$\mathcal{B}(w,v) = \mathcal{B}_a(w,v) + \mathcal{B}_{\mathbf{b}}(w,v) - \mathcal{B}_f(v,w) - \mathcal{B}_f(w,v) + \mathcal{B}_\vartheta(w,v)\,,$$

where

$$\begin{aligned}
\mathcal{B}_a(w,v) &= \sum_{\kappa\in\mathcal{T}_h} \int_\kappa a\nabla w\cdot\nabla v\,\mathrm{d}\mathbf{x}\,,\\
\mathcal{B}_{\mathbf{b}}(w,v) &= \sum_{\kappa\in\mathcal{T}_h} \Big\{ -\int_\kappa (w\,\mathbf{b}\cdot\nabla v - cwv)\,\mathrm{d}\mathbf{x}\\
&\qquad + \int_{\partial_+\kappa} (\mathbf{b}\cdot\mathbf{n}_\kappa)\,w^+v^+\,\mathrm{d}s + \int_{\partial_-\kappa\setminus\Gamma} (\mathbf{b}\cdot\mathbf{n}_\kappa)\,w^-v^+\,\mathrm{d}s \Big\}\,,\\
\mathcal{B}_f(w,v) &= \int_{\Gamma_{\mathcal{I}}\cup\Gamma_{\rm D}} \{\!\{a\nabla_h w\}\!\}\cdot[\![v]\!]\,\mathrm{d}s\,, \qquad \mathcal{B}_\vartheta(w,v) = \int_{\Gamma_{\mathcal{I}}\cup\Gamma_{\rm D}} \vartheta[\![w]\!]\cdot[\![v]\!]\,\mathrm{d}s\,,
\end{aligned}$$

and the linear functional $\ell(\cdot)$ is given by

$$\begin{aligned}
\ell(v) &= \sum_{\kappa\in\mathcal{T}_h} \Big(\int_\kappa fv\,\mathrm{d}\mathbf{x} - \int_{\partial_-\kappa\cap(\Gamma_{\rm D}\cup\Gamma_-)} (\mathbf{b}\cdot\mathbf{n}_\kappa)\,g_{\rm D}\,v^+\,\mathrm{d}s\\
&\quad - \int_{\partial\kappa\cap\Gamma_{\rm D}} g_{\rm D}((a\nabla v^+)\cdot\mathbf{n}_\kappa)\,\mathrm{d}s + \int_{\partial\kappa\cap\Gamma_{\rm N}} g_{\rm N}v^+\,\mathrm{d}s + \int_{\partial\kappa\cap\Gamma_{\rm D}} \vartheta g_{\rm D}v^+\,\mathrm{d}s\Big)\,.
\end{aligned}$$

The discontinuity-penalization parameter ϑ is defined by $\vartheta|_f = \vartheta_f$ for $f \subset \Gamma_{\mathcal{I}}\cup\Gamma_{\rm D}$, where ϑ_f is a nonnegative constant on face f. The precise choice of ϑ_f, which depends on a and the discretization parameters, will be discussed in detail in the next section.

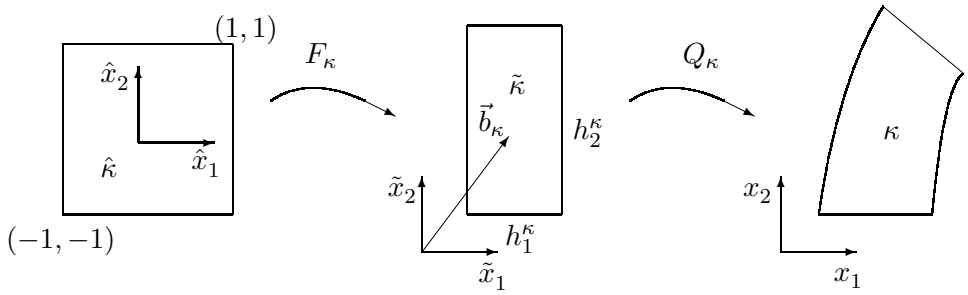


Figure 41: Construction of elements via composition of affine maps and diffeomorphisms.

6.1.2 Stability analysis

In this section we will analyze the DG method (162) in two settings. In the first case, we assume that the polynomial degrees are restricted so that they are isotropic on each element, i.e., for all elements $\kappa \in \mathcal{T}_h$, $\vec{p}_\kappa \equiv p_\kappa$, where $p_\kappa \geq 1$ is an integer; in this case $u_h \in V_{h,\mathbf{p}_{\rm iso}}$. In the second case we admit anisotropic polynomial degrees, but restrict each element $\kappa \in \mathcal{T}_h$ to being an axiparallel image of the unit hypercube (up to a C^1-diffeomorphism). For simplicity of presentation, in this latter case, we assume that $d = 2$; however, we note that all of the results presented in this work naturally generalise to the case $d = 3$, by exploiting analogous arguments to those presented in the sequel. Thereby, in this latter setting, we have that F_κ is an affine mapping of the form

$$F_\kappa(\hat{\mathbf{x}}) = A_\kappa \hat{\mathbf{x}} + \mathbf{b}_\kappa,$$

where $A_\kappa := \frac{1}{2}\mathrm{diag}(h_1^\kappa, h_2^\kappa)$, with h_1^κ and h_2^κ the lengths of the edges of $\tilde{\kappa}$ parallel to the $\tilde{x}_1$- and $\tilde{x}_2$-axes, respectively, $\mathbf{b}_\kappa$ is a two-component real-valued vector and Q_κ is a smooth diffeomorphism as before, see Figure 41.

We now recall the definition of the function $\mathtt{h}$ in $L_\infty(\Gamma_\mathcal{I} \cup \Gamma_{\rm D})$, as $\mathtt{h}(\mathbf{x}) = \min\{m_{\kappa_1}, m_{\kappa_2}\}/m_f$, if $\mathbf{x}$ is in the interior of $f = \partial\kappa_1 \cap \partial\kappa_2$ for two neighboring elements in the mesh $\mathcal{T}_h$, and $\mathtt{h}(\mathbf{x}) = m_\kappa/m_f$, if $\mathbf{x}$ is in the interior of $f = \partial\kappa \cap \Gamma_{\rm D}$, cf. Section 5.4. Similarly, we recall the definition of the function $\mathtt{a}$ in $L_\infty(\Gamma_\mathcal{I} \cup \Gamma_{\rm D})$ by $\mathtt{a}(\mathbf{x}) = \max\{\bar{a}_{\kappa_1}, \bar{a}_{\kappa_2}\}$ if $\mathbf{x}$ is in the interior of $f = \partial\kappa_1 \cap \partial\kappa_2$, and $\mathtt{a}(\mathbf{x}) = \bar{a}_\kappa$ if $\mathbf{x}$ is in the interior of $\partial\kappa \cap \Gamma_{\rm D}$.

In the case when the composite polynomial degree vector $\vec{\mathbf{p}}$ is isotropic, i.e., when $u_h \in V_{h,\mathbf{p}_{\rm iso}}$, where $\mathbf{p}_{\rm iso} = (p_\kappa : \ \kappa \in \mathcal{T}_h)$, we introduce the function $\mathtt{p}(\mathbf{x}) \in L_\infty(\Gamma_\mathcal{I} \cup \Gamma_{\rm D})$ by $\mathtt{p}(\mathbf{x}) = \max\{p_{\kappa_1}, p_{\kappa_2}\}$ if $\mathbf{x}$ is in the interior of $f = \partial\kappa_1 \cap \partial\kappa_2$, and $\mathtt{p}(\mathbf{x}) = p_\kappa$ if $\mathbf{x}$ is in the interior of $\partial\kappa \cap \Gamma_{\rm D}$.

With this notation, we now provide the following coercivity result for the bilinear form $\mathcal{B}(\cdot,\cdot)$ over $V_{h,\mathbf{p}_{\rm iso}} \times V_{h,\mathbf{p}_{\rm iso}}$.

Theorem 6.2 *If ϑ is defined as*

$$\vartheta|_f \equiv \vartheta_f = C_\vartheta \frac{\mathtt{ap}^2}{\mathtt{h}} \quad \textit{for} \quad f \subset \Gamma_\mathcal{I} \cup \Gamma_{\rm D}, \tag{163}$$

then there exists a positive constant C, which depends only on the dimension d, such that

$$\mathcal{B}(v,v) \geq C|||v|||^2 \quad \forall v \in V_{h,\mathbf{p}_{\rm iso}},$$

provided that the constant C_ϑ is chosen such that:

$$C_\vartheta > C'_\vartheta > 0,$$

where C'_ϑ is a sufficiently large positive constant.

Proof: The proof follows in an analogous fashion to Theorem 5.6; see [49] for details. □

We end this section by establishing the coercivity of the bilinear form $\mathcal{B}(\cdot,\cdot)$ over $V_{h,\vec{\mathbf{p}}} \times V_{h,\vec{\mathbf{p}}}$, assuming for simplicity that $d = 2$ and that each element $\kappa \in \mathcal{T}_h$ is an axiparallel image of the unit hypercube (up to a C^1-diffeomorphism).

In this setting the mesh function $\mathtt{h}$ in $L_\infty(\Gamma_\mathcal{I} \cup \Gamma_\mathrm{D})$ defined above, may be equivalently written as $\mathtt{h}(\mathbf{x}) = \min\{h_j^\kappa, h_j^{\kappa'}\}$, if $\mathbf{x}$ is in the interior of $f = \partial\kappa \cap \partial\kappa'$ for two neighboring elements κ, κ' in the mesh $\mathcal{T}_h$, and $\tilde{f} = Q_\kappa^{-1}(f)$ is parallel to the $\tilde{x}_i$–axis, $i, j = 1, 2$, $i \neq j$; $\mathtt{h}(\mathbf{x}) = h_j^\kappa$, if $\mathbf{x}$ is in the interior of $f = \partial\kappa \cap \Gamma_\mathrm{D}$ and $\tilde{f} = Q_\kappa^{-1}(f)$ is parallel to the $\tilde{x}_i$–axis, $i, j = 1, 2$, $i \neq j$. In a similar fashion, we define $\mathtt{p}_a$ in $L_\infty(\Gamma_\mathcal{I} \cup \Gamma_\mathrm{D})$ by $\mathtt{p}_a(\mathbf{x}) = \max\{p_{\kappa,j}, p_{\kappa',j}\}$ for κ, κ' as above; $\mathtt{p}_a(\mathbf{x}) = p_{\kappa,j}$ if $\mathbf{x}$ is in the interior of a boundary face as above. In this case coercivity of $\mathcal{B}(\cdot,\cdot)$ over $V_{h,\vec{\mathbf{p}}} \times V_{h,\vec{\mathbf{p}}}$ can again be shown.

Theorem 6.3 *For a mesh $\mathcal{T}_h$ consisting only of axiparallel images of the unit square (up to a C^1-diffeomorphism), if ϑ is defined as*

$$\vartheta|_f \equiv \vartheta_f = C_\vartheta \frac{\mathtt{ap}_a^2}{\mathtt{h}} \quad \textit{for} \quad f \subset \Gamma_\mathcal{I} \cup \Gamma_\mathrm{D}, \tag{164}$$

then there exists a positive constant C, which depends only on the dimension d, such that

$$\mathcal{B}(v, v) \geq C|||v|||^2 \quad \forall v \in V_{h,\vec{\mathbf{p}}},$$

provided that the constant C_ϑ is chosen such that:

$$C_\vartheta > C'_\vartheta > 0,$$

where C'_ϑ is a sufficiently large positive constant.

Proof: See Georgoulis [41]. □

Remark 6.4 *Theorem 6.3 implies that the direction perpendicular to the face of interest is the important one for ensuring stability. Indeed, in the case of anisotropic diffusion, it is also the case that only diffusion perpendicular to the face need be considered, see Georgoulis [41].*

6.2 *hp*-Error bounds on the hypercube

In this section we now consider the generalization of the approximation results stated in Section 5.5 in the *h*–version case, to the *hp*–setting. As above, we first consider the case when isotropic polynomials are employed on general finite element meshes consisting of tensor product elements, i.e., hypercubes, before dealing with the situation where anisotropic polynomial degrees are exploited.

6.2.1 Isotropic polynomials degrees

In this section, we first consider the case when the local elemental polynomial degree vector $\vec{p}_\kappa$ is constant, i.e., when $p_{\kappa,1} = p_{\kappa,2} = \ldots = p_{\kappa,d}$ for all κ in $\mathcal{T}_h$. To this end, we recall the following notation for the orthogonal L_2–projection operator introduced in Section 5.5. On the reference element $\hat{\kappa}$, we define $\hat{\Pi}_p$ to denote the orthogonal projector in $L_2(\hat{\kappa})$ onto the space of polynomials $\mathcal{Q}_p(\hat{\kappa})$; i.e., given that $\hat{v} \in L_2(\hat{\kappa})$, we define $\hat{\Pi}_p\hat{v}$ by

$$(\hat{v} - \hat{\Pi}_p\hat{v}, \hat{w})_{\hat{\kappa}} = 0$$

for all $\hat{w} \in \mathcal{Q}_p(\hat{\kappa})$, where $(\cdot,\cdot)_{\hat{\kappa}}$ denotes the $L_2(\hat{\kappa})$ inner product. Similarly, we define the L_2-projection operators $\tilde{\Pi}_p$ and Π_p on $\tilde{\kappa}$ and κ, respectively, by the relations

$$\tilde{\Pi}_p\tilde{v} := (\hat{\Pi}_p(\tilde{v} \circ F_\kappa)) \circ F_\kappa^{-1}, \qquad \Pi_p v := (\tilde{\Pi}_p(v \circ Q_\kappa)) \circ Q_\kappa^{-1},$$

for $\tilde{v} \in L_2(\tilde{\kappa})$ and $v \in L_2(\kappa)$, respectively.

With this notation, we now quote the *hp*–analogue of Lemma 5.9.

Lemma 6.5 *Let $\hat{\kappa}$ be the unit d-hypercube, and let $\hat{f}$ denote one of its faces. Given a function $\hat{v} \in H^k(\hat{\kappa})$, the following error bounds hold*

$$\begin{aligned}
\|\hat{v} - \hat{\Pi}_p\hat{v}\|_{L_2(\hat{\kappa})} &\leq \frac{C}{p^s}|\hat{v}|_{H^s(\hat{\kappa})}, && 0 \leq s \leq \min(p+1,k),\\
|\hat{v} - \hat{\Pi}_p\hat{v}|_{H^1(\hat{\kappa})} &\leq \frac{C}{p^{s-3/2}}|\hat{v}|_{H^s(\hat{\kappa})}, && 1 \leq s \leq \min(p+1,k),\\
\|\hat{v} - \hat{\Pi}_p\hat{v}\|_{L_2(\hat{f})} &\leq \frac{C}{p^{s-1/2}}|\hat{v}|_{H^s(\hat{\kappa})}, && 1 \leq s \leq \min(p+1,k),\\
|\hat{v} - \hat{\Pi}_p\hat{v}|_{H^1(\hat{f})} &\leq \frac{C}{p^{s-5/2}}|\hat{v}|_{H^s(\hat{\kappa})}, && 2 \leq s \leq \min(p+1,k),
\end{aligned}$$

where C is a constant dependent only on the dimension d.

Proof: A proof can be found in Houston, Schwab and Süli, [74]. □

Rescaling Lemma 6.5 to the physical element we easily attain the following result, cf. Theorem 5.20 in the *h*–version setting.

Lemma 6.6 *Using the notation of Lemma 5.9, there exists a positive constant C, which*

depends only on the dimension d such that:

$$\|v-\Pi_p v\|_{L_2(\kappa)} \leq \frac{C}{p^s}\left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v},\Sigma_\kappa,U_\kappa)\,\mathrm{d}\tilde{\mathbf{x}}\right]^{\frac{1}{2}},\quad 0\leq s\leq \min(p+1,k), \tag{165}$$

$$|v-\Pi_p v|_{H^1(\kappa)} \leq \frac{C}{p^{s-3/2}}|\sigma_{d,\kappa}|^{-1} \times\left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v},\Sigma_\kappa,U_\kappa)\,\mathrm{d}\tilde{\mathbf{x}}\right]^{\frac{1}{2}},\quad 1\leq s\leq \min(p+1,k), \tag{166}$$

$$\|v-\Pi_p v\|_{L_2(f)} \leq \frac{C}{p^{s-1/2}}|\sigma_{d,\kappa}|^{-1/2} \times\left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v},\Sigma_\kappa,U_\kappa)\,\mathrm{d}\tilde{\mathbf{x}}\right]^{\frac{1}{2}},\quad 1\leq s\leq \min(p+1,k), \tag{167}$$

$$|v-\Pi_p v|_{H^1(f)} \leq \frac{C}{p^{s-5/2}}\left|\frac{m_f}{m_\kappa}\right|^{\frac{1}{2}}|\sigma_{d,\kappa}|^{-1} \times\left[\int_{\tilde{\kappa}} D^s_{\tilde{\kappa}}(\tilde{v},\Sigma_\kappa,U_\kappa)\,\mathrm{d}\tilde{\mathbf{x}}\right]^{\frac{1}{2}},\ 2\leq s\leq \min(p+1,k). \tag{168}$$

Proof: The same arguments from Corollary 5.10 and Lemma 5.16 can be applied to the results from Lemma 6.5 in order to achieve the results. □

Remark 6.7 *Considering once again isotropic elements and bearing in mind Remark 5.19, we see that Lemma 6.6 shares exactly the same convergence results in terms of both h_κ and p as Lemma 4.3 of [50]. Indeed, all four results from Lemma 6.6 show h-optimal convergence rates, with the errors in the L_2–norm exhibiting p-optimal convergence; however, both H^1–bounds are p–suboptimal, with (166) and (168) suboptimal by $p^{1/2}$ and p, respectively. hp–optimal convergence rates have been shown for alternative projection operators, see for example Georgoulis [41], however, as noted in Section 5.5, the L_2–projector is required in the sequel in order to derive a priori bounds for the error in the computed target functional.*

6.2.2 Anisotropic polynomial degrees

In this section we consider the 2–dimensional axiparallel setting, where anisotropic polynomial degrees are admissible. Before we embark with the error analysis, we present some results taken from [42] regarding the approximation error of the orthogonal L_2-projection operator onto the finite element space $V_{h,\vec{\mathbf{p}}}$. All the proofs of the following results can be found in [42].

Let $\hat{u}\in L_2(\hat{I})$, with $\hat{I}\equiv(-1,1)$. We define the L_2-orthogonal projector $\hat{\pi}_p$ on $\hat{I}$ in a standard fashion by means of truncated Legendre series (see, e.g., [99]). With this definition, for $\hat{\kappa}\equiv(-1,1)^2$ we write $\hat{\Pi}_{\vec{p}}: L_2(\hat{\kappa})\rightarrow\mathcal{Q}_{\vec{p}}(\hat{\kappa})$, with composite polynomial degree vector $\vec{p}=(p_1,p_2)$, by

$$\hat{\Pi}_{\vec{p}} := (\hat{\pi}^1_{p_1}\otimes I)(I\otimes\hat{\pi}^2_{p_2}),$$

where $\hat{\pi}^1_{p_1}$ and $\hat{\pi}^2_{p_2}$ denote the one-dimensional L_2-projection operators defined above, with the superscripts $1,2$ indicating the directions in which the one-dimensional projectors are applied, respectively, and $\otimes$ the standard functional tensor product.

Definition 6.8 *Let $\tilde{u} : \tilde{\kappa} \to \mathbb{R}$ and $u : \kappa \to \mathbb{R}$ and assume that there exist mappings $F_\kappa : \hat{\kappa} \to \tilde{\kappa}$, $Q_\kappa : \tilde{\kappa} \to \kappa$ as above. We define the L_2-projection operator $\tilde{\Pi}_{\vec{p}}$ on $\tilde{\kappa}$, with $\vec{p} = (p_1, p_2)$ being the composite polynomial degree vector, by the relation*

$$\tilde{\Pi}_{\vec{p}}\tilde{u} := (\hat{\Pi}_{\vec{p}}(\tilde{u} \circ F_\kappa)) \circ F_\kappa^{-1}, \quad \text{for } \tilde{u} \in L_2(\tilde{\kappa}),$$

where, as before, $\hat{\Pi}_{\vec{p}}$ denotes the L_2-orthogonal projection onto the reference element $\hat{\kappa}$. Moreover, we define the L_2-orthogonal projection operator $\Pi_{\vec{p}}$ on κ, with $\vec{p} = (p_1, p_2)$, by

$$\Pi_{\vec{p}}u := (\tilde{\Pi}_{\vec{p}}(u \circ Q_\kappa)) \circ Q_\kappa^{-1}, \quad \text{for } u \in L_2(\kappa).$$

We introduce some notation which we shall use in the approximation estimates below. We define

$$\Phi(p, s, h) := \left(\frac{(p-(s-1))!}{(p+(s-1))!}\right)\left(\frac{h}{2}\right)^{2(s-1)}, \tag{169}$$

where p and s are integers such that $1 \le s \le p$. Let $J_{Q_\kappa} = \big((J_{Q_\kappa})_{ij}\big)_{i,j=1,2}$ denote the Jacobi matrix of Q_κ, which is assumed to be a (smooth) diffeomorphism. In the following approximation estimates for the L_2-projection error, the generic non-negative constants C_κ, C_κ^1, and C_κ^2, $\kappa \in \mathcal{T}_h$, are assumed to be dependent on Q_κ but not on the elemental polynomial degree or the affine map F_κ. Moreover, we assume that C_κ^1 and C_κ^2, $\kappa \in \mathcal{T}_h$, are of the form

$$C_\kappa^1 := \begin{cases} 1, & \text{if } Q_\kappa = \mathrm{id}, \\ C(J_{Q_\kappa}), & \text{otherwise}, \end{cases}$$

$$C_\kappa^2 := \begin{cases} 0, & \text{if } Q_\kappa = \mathrm{id}, \\ C(J_{Q_\kappa}), & \text{otherwise}, \end{cases}$$

where $C(J_{Q_\kappa})$ is a generic positive constant depending on J_{Q_κ} only. Finally, we define $\partial\hat{\kappa}_1 := (-1,1) \times \{\pm 1\}$, $\partial\hat{\kappa}_2 := \{\pm 1\} \times (-1,1)$, $\partial\tilde{\kappa}_i := F_\kappa(\partial\hat{\kappa}_i)$ and $\partial\kappa_i := Q_\kappa(\partial\tilde{\kappa}_i)$, for $i = 1, 2$.

The following interpolation estimates then hold.

Lemma 6.9 *Let $u \in H^k(\kappa)$, for $k \ge 2$; then, for $\tilde{u} := u \circ Q_\kappa$, $\vec{p} = (p_1, p_2)$ and $p_1, p_2 \ge 1$, we have*

$$\|u - \Pi_{\vec{p}}u\|^2_{L_2(\kappa)} \le C_\kappa M^0_\kappa, \tag{170}$$

where

$$M^0_\kappa := \sum_{i=1}^{2} \Phi(p_i, s_i, h_i)\left(\frac{h_i}{2p_i}\right)^2 \|\tilde{\partial}_i^{s_i}\tilde{u}\|^2_{L_2(\tilde{\kappa})}, \tag{171}$$

and

$$\|\partial_i(u - \Pi_{\vec{p}}u)\|^2_{L_2(\kappa)} \le C^1_\kappa M^1_{\kappa,i} + C^2_\kappa M^1_{\kappa,j}, \tag{172}$$

with

$$M^1_{\kappa,i} := p_i\Phi(p_i, s_i, h_i)\|\tilde{\partial}_i^{s_i}\tilde{u}\|^2_{L_2(\tilde{\kappa})} + \Phi(p_j, s_j, h_j)\|\tilde{\partial}_j^{s_j-1}\tilde{\partial}_i\tilde{u}\|^2_{L_2(\tilde{\kappa})}, \tag{173}$$

where $i, j = 1, 2$, $i \ne j$, $1 \le s_i \le \min\{p_i + 1, k\}$, for $i = 1, 2$, and $\tilde{\partial}_i$ is the partial derivative in the $\tilde{x}_i$-direction in the $\tilde{x}_1\tilde{x}_2$-plane.

Lemma 6.10 *Let $u \in H^k(\kappa)$, with $k \geq 1$; then we have*

$$\|u - \Pi_{\vec{p}} u\|^2_{L_2(\partial\kappa_i)} \leq C_\kappa M^0_{\partial\kappa,i}, \tag{174}$$

with

$$\begin{aligned} M^0_{\partial\kappa,i} &:= \Phi(p_j, s_j, h_j)\frac{h_j}{2p_j}\|\tilde{\partial}_j^{s_j}\tilde{u}\|^2_{L_2(\tilde{\kappa})} + \Phi(p_i, s_i, h_i)\frac{h_i}{h_j}\frac{h_i}{2p_i}\|\tilde{\partial}_i^{s_i}\tilde{u}\|^2_{L_2(\tilde{\kappa})} \\ &\quad + \Big(\frac{p_j}{p_i} + 1\Big)\Phi(p_i, s_i, h_i)\frac{h_j}{2p_j}\|\tilde{\partial}_i^{s_i-1}\tilde{\partial}_j\tilde{u}\|^2_{L_2(\tilde{\kappa})}, \end{aligned}$$

with $i, j = 1, 2$, $i \neq j$, $1 \leq s_i \leq \min\{p_i + 1, k\}$, and $p_i \geq 1$, for $i = 1, 2$.

Lemma 6.11 *Let $u \in H^k(\kappa)$, with $k \geq 2$; then the following error estimates hold:*

$$\|\partial_i(u - \Pi_{\vec{p}} u)\|^2_{L_2(\partial\kappa_i)} \leq C^1_\kappa M^1_{\partial\kappa,i} + C^2_\kappa M^2_{\partial\kappa,i}, \tag{175}$$

$$\|\partial_j(u - \Pi_{\vec{p}} u)\|^2_{L_2(\partial\kappa_i)} \leq C^1_\kappa M^2_{\partial\kappa,i} + C^2_\kappa M^1_{\partial\kappa,i}, \tag{176}$$

with

$$\begin{aligned} M^1_{\partial\kappa,i} &:= \Phi(p_i, s_i, h_i)\frac{2p_i}{h_i}\left(p_i\frac{h_i}{h_j}\|\tilde{\partial}_i^{s_i}\tilde{u}\|^2_{L_2(\tilde{\kappa})} + \Big(1 + \frac{p_i}{p_j}\Big)\frac{h_j}{h_i}\|\tilde{\partial}_i^{s_i-1}\tilde{\partial}_j\tilde{u}\|^2_{L_2(\tilde{\kappa})}\right) \\ &\quad + \Phi(p_j, s_j, h_j)\frac{2p_j}{h_j}\|\tilde{\partial}_j^{s_j-1}\tilde{\partial}_i\tilde{u}\|^2_{L_2(\tilde{\kappa})}, \end{aligned} \tag{177}$$

for $i, j = 1, 2$, $i \neq j$, $1 \leq s_i \leq \min\{p_i + 1, k\}$, $p_i \geq 1$, $i = 1, 2$, and

$$M^2_{\partial\kappa,i} := p_j^2\Phi(p_j, s_j, h_j)\frac{2p_j}{h_j}\|\tilde{\partial}_j^{s_j}\tilde{u}\|^2_{L_2(\tilde{\kappa})} + p_j\Phi(p_i, s_i, h_i)\frac{2p_j}{h_j}\|\tilde{\partial}_i^{s_i-1}\tilde{\partial}_j\tilde{u}\|^2_{L_2(\tilde{\kappa})}, \tag{178}$$

for $2 \leq s_i \leq \min\{p_i + 1, k\}$.

Proof: For each of the lemmata 6.9-6.11 full proofs can be found in Georgoulis [42]. In each case, the idea is to split up the L_2-projection operator on the reference element into a tensor-product composition of one-dimensional L_2-projectors and apply one dimensional results (for example, see Schwab [99]); scaling back to the physical element then completes the proof. □

Remark 6.12 *By using* Stirling's formula

$$n! \sim \sqrt{2\pi}n^{n+1/2}\mathrm{e}^{-n}, \quad n > 0, \tag{179}$$

we see that for $p \geq 1$,

$$\Phi(p, s, h) \leq C(s)p^{-2(s-1)}h^{2(s-1)}. \tag{180}$$

Thus, if we consider isotropic polynomial degrees in the results from the Lemmata 6.9-6.11 and apply (180) we return to the same asymptotic results in terms of p as for Lemma 6.6. Considering also isotropic h, that is $h_1^\kappa \sim h_2^\kappa$, then we recover the same approximation results from Harriman et al. [50].

We shall now consider the case where the functions we are approximating are analytic. In this case we shall see that the L_2 projection provides p–exponential convergence, a very desirable property, which improves on the merely algebraic convergence in p witnessed above. To this end, we state the following result from [49, 41].

Lemma 6.13 *Let $u : \kappa \to \mathbb{R}$ have an analytic extension to an open neighbourhood of $\bar{\kappa}$. Also, let p, s, and n be positive integers such that*

$$0 \leq n \leq s := \alpha p + n \leq p,$$

with $0 < \alpha < 1$. Then the following bounds hold

$$\begin{aligned} \Phi(p, s+1, h)\|\partial_i^{s+1}\partial_j^m u\|^2_{L_2(\kappa)} &\leq C_u h^{2s} p^{\min\{3, n+\frac{5}{2}\}} \mathrm{e}^{-rp}|\kappa|, \\ \Phi(p, s+1, h)\|\partial_i^{s}\partial_j^m u\|^2_{L_2(\kappa)} &\leq C_u h^{2s} p^{\min\{3, n+\frac{5}{2}\}} \mathrm{e}^{-rp}|\kappa|, \end{aligned}$$

where $m \in \{0, 1\}$ and r, $C_u > 0$ are constants that depend on n and u, with $i, j \in \{1, 2\}$ for $i \neq j$, and $|\kappa|$ denotes the Lebesgue measure of the domain κ.

We notice that the results of Lemmas 6.9-6.11 all include terms of the form

$$\Phi(p, s, h)\|\partial_i^{s-1} u\|^2_{L_2(\kappa)} \text{ or } \Phi(p, s, h)\|\partial_i^{s}\partial_j u\|^2_{L_2(\kappa)},$$

and hence Lemma 6.13 can be used to show that, for an analytic function u, the L_2-projector achieves p-exponential convergence in both the L_2-norm and H^1 semi-norm on the element and the element boundary.

6.3 *A priori* error analysis

On the basis of the approximation results developed within the previous section, we now proceed to derive *a priori* error bounds for general linear target functionals $J(\cdot)$ of the solution. To this end, we first consider the case when isotropic polynomial degrees are employed, i.e., when $u_h \in V_{h,\mathbf{p}_{\text{iso}}}$. In this case, along with the assumption that the element volumes satisfy the bounded local variation condition (147), we also assume bounded local variation of the polynomial degrees, i.e., there exists a constant $C_9 > 1$, such that for any pair of elements κ and κ' sharing a (d–1)-dimensional face

$$C_9^{-1} < p_\kappa / p_{\kappa'} < C_9. \tag{181}$$

Theorem 6.14 *Let $\Omega \subset \mathbb{R}^d$ be a bounded polyhedral domain, $\mathcal{T}_h = \{\kappa\}$ a subdivision of Ω, such that the elemental volumes and polynomial degrees satisfy the bounded local variation conditions (147) and (181), respectively. Then, assuming that conditions (161), (117), and (129) on the data hold, and $u|_\kappa \in H^{k_\kappa}(\kappa)$, $k_\kappa \geq 2$, for $\kappa \in \mathcal{T}_h$, and $z|_\kappa \in H^{l_\kappa}(\kappa)$, $l_\kappa \geq 2$, for $\kappa \in \mathcal{T}_h$, then the solution $u_h \in V_{h,\mathbf{p}_{\text{iso}}}$ of (162) obeys the error bound*

$$\begin{aligned} &|J(u) - J(u_h)|^2 \\ &\quad \leq C \left(\sum_{\kappa \in \mathcal{T}_h} \frac{1}{\sigma_{d,\kappa}^2} \left\{ \frac{\alpha}{p_\kappa^{2(s_\kappa - 3/2)}} + \frac{\beta_2 \sigma_{d,\kappa}}{p_\kappa^{2(s_\kappa - 1/2)}} + \frac{(\beta_1 + \gamma_1)\sigma_{d,\kappa}^2}{p_\kappa^{2s_\kappa}} \right\} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^{s_\kappa}(\tilde{u}, \Sigma_\kappa, U_\kappa)\, \mathrm{d}\tilde{\mathbf{x}} \right) \\ &\qquad \times \left(\sum_{\kappa \in \mathcal{T}_h} \frac{1}{\sigma_{d,\kappa}^2} \left\{ \frac{\alpha}{p_\kappa^{2(t_\kappa - 3/2)}} + \frac{\beta_2 \sigma_{d,\kappa}}{p_\kappa^{2(t_\kappa - 1)}} + \frac{(\beta_1 + \gamma_2)\sigma_{d,\kappa}^2}{p_\kappa^{2t_\kappa}} \right\} \int_{\tilde{\kappa}} D_{\tilde{\kappa}}^{t_\kappa}(\tilde{z}, \Sigma_\kappa, U_\kappa)\, \mathrm{d}\tilde{\mathbf{x}} \right), \end{aligned}$$

for $2 \leq s_\kappa \leq \min(p_\kappa + 1, k_\kappa)$ *and* $2 \leq t_\kappa \leq \min(p_\kappa + 1, l_\kappa)$, *where* $\alpha|_\kappa = \bar{a}_{\tilde{\kappa}}$, $\beta_1|_\kappa = \|c + \nabla \cdot \mathbf{b}\|_{L_\infty(\kappa)}$, $\beta_2|_\kappa = \|\mathbf{b}\|_{L_\infty(\kappa)}$, $\gamma_1|_\kappa = \|c/c_0\|^2_{L_\infty(\kappa)}$, $\gamma_2|_\kappa = \|(c + \nabla \cdot \mathbf{b})/c_0\|^2_{L_\infty(\kappa)}$, *for all* $\kappa \in \mathcal{T}_h$. *Here,* C *is a constant depending on the dimension* d *and the parameters* C_i, $i = 1, \ldots, 9$.

Proof: The proof is analogous to that for Theorem 5.23; however, here, we pick $\epsilon_\kappa = p_\kappa^2/\sigma_{d,\kappa}$ and use the interpolation results from Lemma 6.6 together with the discontinuity penalisation term defined by

$$\vartheta|_f = C_\vartheta \frac{\mathtt{ap}^2}{\mathtt{h}}.$$

□

Remark 6.15 *Let us now discuss some special cases of the general error bound derived in Theorem 6.14. To this end, for simplicity, we assume uniform orders* $p_\kappa = p$, $s_\kappa = s$, $t_k = t$, $k_\kappa = k$, $l_\kappa = l$, s, t, k, l *integers for all* $\kappa \in \mathcal{T}_h$, *and uniform isotropic elements with mesh size* h. *In the diffusion dominated case, Theorem 6.14 indicates that the error in computed target functional may be bounded as follows*

$$\begin{aligned} |J(u) - J(u_h)| &\leq C \frac{h^{s+t-2}}{p^{s+t-2}} p\, |u|_{H^s(\Omega)} |z|_{H^t(\Omega)} && (182) \\ &\leq C \frac{h^{s+t-2}}{p^{k+l-2}} p\, \|u\|_{H^k(\Omega)} \|z\|_{H^l(\Omega)}, && (183) \end{aligned}$$

where $2 \leq s \leq \min(p + 1, k)$ *and* $2 \leq t \leq \min(p + 1, l)$. *We note that in the transition from (182) to (183) the generic constant* C *has increased by a factor of* $(k-1)^{k-2}(l-1)^{l-2}$. *This error bound is optimal with respect to* h *but suboptimal in* p *by one order, cf. [50]. For the strictly hyperbolic case* ($a \equiv 0$), *the error bound in Theorem 6.14 becomes*

$$\begin{aligned} |J(u) - J(u_h)| &\leq C \frac{h^{s+t-1}}{p^{s+t-1}} p^{1/2} |u|_{H^s(\Omega)} |z|_{H^t(\Omega)} \\ &\leq C \frac{h^{s+t-1}}{p^{k+l-1}} p^{1/2} \|u\|_{H^k(\Omega)} \|z\|_{H^l(\Omega)}. \end{aligned}$$

This bound is once again optimal in h, *but suboptimal in* p *by* $p^{1/2}$, *cf. [77].*

Once again we return to the case of axiparallel elements; in this setting, we consider a slight variation concerning the bounded local variation conditions on the element sizes and polynomial degrees which we assumed for the proof of Theorem 6.14. Indeed, here we now assume that there exist ρ_i and δ_i, for $i = 1, 2$, such that

$$\rho_i^{-1} \leq p_i^\kappa / p_i^{\kappa'} \leq \rho_i, \tag{184}$$

$$\delta_i^{-1} \leq h_i^\kappa / h_i^{\kappa'} \leq \delta_i, \tag{185}$$

$i = 1, 2$, for all pairs of neighbouring elements κ and κ'.

Theorem 6.16 *Let* $\Omega \subset \mathbb{R}^2$ *be an axiparallel polygonal domain,* $\mathcal{T}_h = \{\kappa\}$ *a subdivision of* Ω *into axiparallel images of the 2-hypercube, such that the bounded local variation conditions,*

(184) and (185), hold. Then, assuming that conditions (161), (117), and (129) on the data hold, and $u|_\kappa \in H^{k_\kappa}(\kappa)$, $k_\kappa \geq 2$, *for* $\kappa \in \mathcal{T}_h$, *and* $z|_\kappa \in H^{l_\kappa}(\kappa)$, $l_\kappa \geq 2$, *for* $\kappa \in \mathcal{T}_h$, *then the solution* $u_h \in V_{h,\vec{\mathbf{p}}}$ *of (162) obeys the error bound*

$$\begin{aligned}
|J(u) - J(u_h)|^2 \;\leq\; & C\Big(\sum_{\kappa\in\mathcal{T}_h} \sum_{i=1}^{2} \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \max_{(m,n)\in A} \Big\{ \Big(\frac{p_j^\kappa}{p_i^\kappa}\Big)^m \Big(\frac{h_i^\kappa}{h_j^\kappa}\Big)^n \Big\} \\
& \times \big(\alpha_\kappa p_i^\kappa + \frac{h_i^\kappa}{p_i^\kappa}\beta_2 + \Big(\frac{h_i^\kappa}{p_i^\kappa}\Big)^2(\beta_1+\gamma_1)\big)|u|^2_{s_i^\kappa,\kappa,i}\Big) \\
& \times \Big(\sum_{\kappa\in\mathcal{T}_h} \sum_{i=1}^{2} \Phi(p_i^\kappa, t_i^\kappa, h_i^\kappa) \max_{(m,n)\in A} \Big\{ \Big(\frac{p_j^\kappa}{p_i^\kappa}\Big)^m \Big(\frac{h_i^\kappa}{h_j^\kappa}\Big)^n \Big\} \\
& \times \big(\alpha_\kappa p_i^\kappa + h_i^\kappa\beta_2 + \Big(\frac{h_i^\kappa}{p_i^\kappa}\Big)^2(\beta_1+\gamma_2)\big)|z|^2_{t_i^\kappa,\kappa,i}\Big),
\end{aligned}$$

with $A = \{(0,0),(0,1),(0,2),(-1,0),(-1,1)(1,2),(2,1),(2,2)\}$, *and*

$$|w|_{r,\kappa,i} := \Big(\|\tilde{\partial}_i^r w\|^2_{L_2(\hat{\kappa})} + \Big(\frac{h_j^\kappa}{h_i^\kappa}\Big)^2 \|\tilde{\partial}_i^{r-1}\tilde{\partial}_j w\|^2_{L_2(\hat{\kappa})}\Big)^{1/2},$$

for $2 \leq s_i^\kappa \leq \min(p_\kappa+1, k_\kappa)$ *and* $2 \leq t_i^\kappa \leq \min(p_\kappa+1, l_\kappa)$, *where* $\alpha|_\kappa = \bar{a}_{\tilde{\kappa}}$, $\beta_1|_\kappa = \|c + \nabla\cdot\mathbf{b}\|_{L_\infty(\kappa)}$, $\beta_2|_\kappa = \|\mathbf{b}\|_{L_\infty(\kappa)}$, $\gamma_1|_\kappa = \|c/c_0\|^2_{L_\infty(\kappa)}$, $\gamma_2|_\kappa = \|(c+\nabla\cdot\mathbf{b})/c_0\|^2_{L_\infty(\kappa)}$, *for all* $\kappa \in \mathcal{T}_h$. *Here,* C *is a constant depending on the parameters* δ_i, ρ_i, $i = 1, 2$.

Proof: Inequality (151) is also applicable in this case, by rearranging the terms we obtain

$$\begin{aligned}
|J(u) - J(u_h)|^2 \;\leq\; & C\left(\sum_{\kappa\in\mathcal{T}_h} \left\{ \bar{a}_{\tilde{\kappa}} \left(\|\nabla\eta\|^2_{L_2(\kappa)} + \frac{\bar{a}_{\tilde{\kappa}}}{\vartheta}\|\nabla\eta\|^2_{L_2(\partial\kappa)} + \frac{\vartheta}{\bar{a}_{\tilde{\kappa}}}\|\eta\|^2_{L_2(\partial\kappa)} \right) \right.\right. \\
& \left.\left. + \beta_2 \left(\epsilon_\kappa^{-1}\|\nabla\eta\|^2_{L_2(\kappa)} + \|\eta\|^2_{L_2(\partial\kappa)} \right) \right) + (\beta_1+\gamma_1)\|\eta\|^2_{L_2(\kappa)} \right\} \Big) \\
& \times \left(\sum_{\kappa\in\mathcal{T}_h} \left\{ \bar{a}_{\tilde{\kappa}} \left(\|\nabla w\|^2_{L_2(\kappa)} + \frac{\bar{a}_{\tilde{\kappa}}}{\vartheta}\|\nabla w\|^2_{L_2(\partial\kappa)} + \frac{\vartheta}{\bar{a}_{\tilde{\kappa}}}\|w\|^2_{L_2(\partial\kappa)} \right) \right.\right. \\
& \left.\left. \beta_2 \left(\epsilon_\kappa\|w\|^2_{L_2(\kappa)} + \|w\|^2_{L_2(\partial\kappa)} \right) + (\beta_1+\gamma_2)\|w\|^2_{L_2(\kappa)} \right\} \right). \\
\equiv\; & C\left(\sum_{\kappa\in\mathcal{T}_h} I^\kappa_{1,\eta} + I^\kappa_2 + I^\kappa_3 \right) \times \left(\sum_{\kappa\in\mathcal{T}_h} I^\kappa_{1,w} + I^\kappa_4 + I^\kappa_5 \right).
\end{aligned}$$

For term $I^\kappa_{1,\eta}$ (similarly $I^\kappa_{1,w}$) we first split into contributions from the faces $\partial\kappa_i$ and $\partial\kappa_j$, such that

$$I^\kappa_{1,\eta} \;\leq\; \bar{a}_{\tilde{\kappa}} \left(\|\nabla\eta\|^2_{L_2(\kappa)} + \sum_{i=1}^{2} \frac{\bar{a}_{\tilde{\kappa}}}{\vartheta}\|\nabla\eta\|^2_{L_2(\partial\kappa_i)} + \frac{\vartheta}{\bar{a}_{\tilde{\kappa}}}\|\eta\|^2_{L_2(\partial\kappa_i)} \right).$$

Then, employing the interpolation result from Lemma 6.9 we obtain

$$\begin{aligned}
\|\nabla\eta\|^2_{L_2(\kappa)} &\leq C\sum_{i=1}^{2}\left(p_i^\kappa\Phi(p_i^\kappa,s_i^\kappa,h_i^\kappa)\|\tilde{\partial}_i^{s_i^\kappa}\tilde{u}\|^2+\Phi(p_i^\kappa,s_i^\kappa,h_i^\kappa)\|\tilde{\partial}_i^{s_i^\kappa-1}\tilde{\partial}_j\tilde{u}\|^2\right)\\
&\leq C\sum_{i=1}^{2}p_i^\kappa\Phi(p_i^\kappa,s_i^\kappa,h_i^\kappa)\left[1+\left(\frac{h_i^\kappa}{h_j^\kappa}\right)^2\right]|u|^2_{s_i^\kappa,\kappa,i}\\
&\leq C\sum_{i=1}^{2}p_i^\kappa\Phi(p_i^\kappa,s_i^\kappa,h_i^\kappa)\max_{n=\{0,2\}}\left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n|u|^2_{s_i^\kappa,\kappa,i}.
\end{aligned}$$

By using the definition of the discontinuity penalization term ϑ from (164), the results of Lemma 6.11 and utilizing the bounded local variation conditions, we also see that

$$\begin{aligned}
&\sum_{i=1}^{2}\frac{\bar{a}_{\bar{\kappa}}}{\vartheta}\|\nabla\eta\|^2_{L_2(\partial\kappa_i)}\\
&\quad\leq C\sum_{i=1}^{2}\Phi(p_i^\kappa,s_i^\kappa,h_i^\kappa)\left(\frac{h_j^\kappa}{(p_j^\kappa)^2}\left[\frac{2(p_i^\kappa)^2}{h_j^\kappa}\|\tilde{\partial}_i^{s_i^\kappa}\tilde{u}\|^2+\left(\frac{p_i^\kappa h_j^\kappa}{h_i^2}\left(1+\frac{p_i^\kappa}{p_j^\kappa}\right)+\frac{(p_j^\kappa)^2}{h_j^\kappa}\right)\|\tilde{\partial}_i^{s_i^\kappa-1}\tilde{\partial}_j\tilde{u}\|^2\right]\right.\\
&\qquad\left.+\frac{h_i^\kappa}{(p_i^\kappa)^2}\left[\frac{(p_i^\kappa)^3}{h_i^\kappa}\|\tilde{\partial}_i^{s_i^\kappa}\tilde{u}\|^2+\frac{p_i^\kappa}{h_i^\kappa}\|\tilde{\partial}_i^{s_i^\kappa-1}\tilde{\partial}_j\tilde{u}\|^2\right]\right)\\
&\quad\leq C\sum_{i=1}^{2}p_i^\kappa\Phi(p_i^\kappa,s_i^\kappa,h_i^\kappa)\left[\left(\frac{p_j^\kappa}{p_i^\kappa}\right)^{-1}+1+\left(\frac{h_i^\kappa}{h_j^\kappa}\right)^2\right]|u|^2_{s_i^\kappa,\kappa,i}\\
&\quad\leq C\sum_{i=1}^{2}p_i^\kappa\Phi(p_i^\kappa,s_i^\kappa,h_i^\kappa)\max_{(m,n)\in A_1}\left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m\left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n|u|^2_{s_i^\kappa,\kappa,i},
\end{aligned}$$

where $A_1=\{(0,0),(0,2),(-1,0)\}$.

Similarly, by using Lemma 6.10 we obtain:

$$\sum_{i=1}^{2}\frac{\vartheta}{\bar{a}_\kappa}\|\eta\|_{L_2(\partial\kappa_i)}\leq C\sum_{i=1}^{2}p_i^\kappa\Phi(p_i^\kappa,s_i^\kappa,h_i^\kappa)\max_{(m,n)\in A_2}\left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m\left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n|u|^2_{s_i^\kappa,\kappa,i},$$

where $A_2=\{(1,2),(2,2)\}$. Hence, it follows that

$$I^\kappa_{1,\eta}\leq C\sum_{i=1}^{2}p_i^\kappa\Phi(p_i^\kappa,s_i^\kappa,h_i^\kappa)\max_{(m,n)\in A}\left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m\left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n|u|^2_{s_i^\kappa,\kappa,i},$$

and

$$I^\kappa_{1,w}\leq C\sum_{i=1}^{2}p_i^\kappa\Phi(p_i^\kappa,t_i,h_i^\kappa)\max_{(m,n)\in A}\left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m\left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n|z|^2_{t_i^\kappa,\kappa,i}.$$

For terms I_2^κ and I_4^κ we make the selection $\epsilon_\kappa = \max_{i=1,2}((p_i^\kappa)^2/h_i^\kappa)$ and using the same techniques as above we achieve:

$$\begin{aligned} I_2^\kappa &\leq \beta_2 \sum_{i=1}^{2} \frac{h_i^\kappa}{p_i^\kappa} \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) \max_{(m,n)\in A_3} \left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m \left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n |u|^2_{s_i^\kappa,\kappa,i}, \\ I_4^\kappa &\leq \beta_2 \sum_{i=1}^{2} h_i^\kappa \Phi(p_i^\kappa, t_i^\kappa, h_i^\kappa) \max_{(m,n)\in A_4} \left(\frac{p_j^\kappa}{p_i^\kappa}\right)^m \left(\frac{h_i^\kappa}{h_j^\kappa}\right)^n |z|^2_{t_i^\kappa,\kappa,i}, \end{aligned}$$

where $A_3 = \{(0,0),(0,1),(-1,1)\}$ and $A_4 = \{(0,0),(0,1),(2,1),(2,2)\}$.

A simple use of Lemma 6.9 also yields

$$\begin{aligned} I_3^\kappa &\leq (\beta_1+\gamma_1) \sum_{i=1}^{2} \left(\frac{h_i^\kappa}{p_i^\kappa}\right)^2 \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa) |u|^2_{s_i^\kappa,\kappa,i}, \\ I_5^\kappa &\leq (\beta_1+\gamma_2) \sum_{i=1}^{2} \left(\frac{h_i^\kappa}{p_i^\kappa}\right)^2 \Phi(p_i^\kappa, t_i^\kappa, h_i^\kappa) |w|^2_{t_i^\kappa,\kappa,i}. \end{aligned}$$

Combining the results for terms I_1-I_5 completes the proof. □

Remark 6.17 *Upon application of Stirling's formula for the factorials arising in the definition of Φ, as in Remark 6.12, it can be shown that the error estimate stated in Theorem 6.16 is h-optimal and slightly p-suboptimal (by one order of p). This is in complete agreement with the results presented for the isotropic case in [50].*

When the analytical solution of both the primal and adjoint problems are sufficiently smooth, then it can be shown that the error converges to zero at an exponential rate with respect to the local (directional) polynomial degrees. More precisely, we state the following result.

Corollary 6.18 *Let $\Omega \subset \mathbb{R}^2$ be a bounded polyhedral domain, $\mathcal{T} = \{\kappa\}$ a 1-irregular subdivision of Ω, such that the mesh parameters satisfy the bounded local variation conditions (184) and (185). Then, assuming that conditions (161), (117), and (129) hold, and that u, z are analytic functions on a neighbourhood of Ω, the solution $u_h \in V_{h,\vec{\mathbf{p}}}$ of (162) obeys the error bound*

$$|J(u) - J(u_h)|^2 \leq C(\alpha,\beta_1,\beta_2,\gamma_1,\gamma_2) \times \Big(\sum_{\kappa\in\mathcal{T}} \sum_{i=1}^{2} \mathrm{e}^{-r_i p_i^\kappa} N_i^\kappa\Big)\Big(\sum_{\kappa\in\mathcal{T}} \sum_{i=1}^{2} \mathrm{e}^{-q_i p_i^\kappa} N_i^\kappa\Big),$$

where

$$N_i^\kappa := \big(h_i^\kappa\big)^{2s_i^\kappa} |\tilde{\kappa}| \max_{(m,n)\in A} \Big\{ \big(p_i^\kappa\big)^{4-m} \big(p_j^\kappa\big)^m \Big(\frac{h_i^\kappa}{h_j^\kappa}\Big)^n \Big\},$$

r_i, q_i are positive constants depending on the domain of analyticity of u and z, respectively, and $|\cdot|$ denotes the two-dimensional Lebesgue measure of a (measurable) subset of Ω; the set A and the data-related constants $\alpha, \beta_1, \beta_2, \gamma_1$, and γ_2 are as in the statement of Theorem 6.16.

Proof: The result follows simply after applying Lemma 6.13 to

$$\Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa)|u|^2_{s_i^\kappa,\kappa,i} = \Phi(p_i^\kappa, s_i^\kappa, h_i^\kappa)\Big(\|\tilde{\partial}_i^{s_i} u\|^2_{L_2(\hat{\kappa})} + \Big(\frac{h_j^\kappa}{h_i^\kappa}\Big)^2 \|\tilde{\partial}_i^{s_i-1}\tilde{\partial}_j u\|^2_{L_2(\hat{\kappa})}\Big),$$

and similarly to

$$\Phi(p_i^\kappa, t_i^\kappa, h_i^\kappa)|z|^2_{t_i^\kappa,\kappa,i} = \Phi(p_i^\kappa, t_i^\kappa, h_i^\kappa)\Big(\|\tilde{\partial}_i^{t_i} z\|^2_{L_2(\hat{\kappa})} + \Big(\frac{h_j^\kappa}{h_i^\kappa}\Big)^2 \|\tilde{\partial}_i^{t_i-1}\tilde{\partial}_j z\|^2_{L_2(\hat{\kappa})}\Big).$$

□

The *a priori* error analysis developed in this section clearly shows that in the case when the Sobolev regularity of the primal solution u, or the adjoint solution z exceed the polynomial degree of the approximating solutions it will be more beneficial to increase the polynomial degree rather than decreasing the size of the mesh. Indeed, in the case when both u and z are real analytic functions polynomial enrichment can lead to exponential convergence. The construction of an automated procedure which is capable of computationally estimating the smoothness of both u and z is the subject of the following section.

6.4 *hp*–Adaptivity on isotropically refined meshes

Recalling the measurement problem stated in Sections 4.1 & 5.7: the aim of the computation is to design an appropriate "optimal" finite element space $V_{h,\vec{\mathbf{p}}}$ such that

$$|J(u) - J(u_h)| \leq \texttt{TOL}\,,$$

where $\texttt{TOL} > 0$ is a given user-defined tolerance. By optimal we mean that the above error control should be attained using a minimal number of degrees of freedom. For simplicity, in this section we first consider the case when both the underlying finite element mesh $\mathcal{T}_h$ and the polynomial distribution are isotropic; thereby, $u_h \in V_{h,\mathbf{p}_{\rm iso}}$. The extension to general anisotropic finite element spaces will be considered in the following section.

Following the discussion presented in Sections 4.1 & 5.7, we exploit the *a posteriori* error bound (154) with z replaced by a suitable numerical approximation, denoted by $\bar{z}_h$. Thereby, in practice we enforce the stopping criterion

$$\sum_{\kappa\in\mathcal{T}_h} |\bar{\eta}_\kappa| \equiv \mathcal{R}_{|\Omega|}(u_h, \bar{z}_h - z_h) \leq \texttt{TOL}\,. \tag{186}$$

If (186) is not satisfied, then the elements are marked for refinement/derefinement according to the size of the (approximate) error indicators $|\bar{\eta}_\kappa|$.

Once an element has been selected for refinement/derefinement the key step in the design of such an (isotropic) *hp*–adaptive algorithm is the local decision taken on each element κ in the computational mesh as to which refinement strategy (i.e., *h*-refinement *via* local mesh subdivision or *p*-refinement by increasing the degree of the local polynomial approximation) should be employed on κ in order to obtain the greatest reduction in the error per unit cost.

The *a priori* estimates developed in the previous section clearly indicate that if either u or z are smooth then a high polynomial degree is preferable to a small mesh size, whereas if u and z are both nonsmooth then a small mesh size should be utilized, cf. [50, 79]. With

this in mind, should an element be selected for refinement and both u and z are nonsmooth, we perform a mesh subdivision, otherwise polynomial enrichment is exploited. Similarly, if an element is flagged for derefinement, then if neither u nor z are smooth we carry out a p-derefinement, else an h-derefinement is undertaken. Of course, since u and z are in general unknown analytically, the local smoothness of these solutions cannot be determined. Motivated by the lack of precise information about the local regularity of the analytical solutions u and z, various algorithms have been developed in the literature with the aim to identify those parts of the computational domain where a given function w, say, may be perceived as being 'smooth' and regions where w is 'non-smooth'. Below we provide a brief review of existing methods which have been developed within the literature.

- *Use of a priori information.* For a linear elliptic boundary–value problem with piecewise analytic coefficients, forcing functions and boundary data, on a computational domain Ω with a piecewise analytic boundary surface $\partial\Omega$, the solution will be an analytic function everywhere, except in the neighbourhood of singularities in the data. Thereby, h–refinement may be employed in those elements in the computational domain whose closures contain such singularities, with p–refinement performed elsewhere. This approach has been employed by Owens and co-workers, for example; cf. [106, 24].

- *Type–parameter.* In this strategy it is assumed that on each element κ in the computational mesh on Ω, one has a local refinement indicator $\eta_\kappa(u_{h,p}, h_\kappa, p_\kappa)$ (not adjoint-based), which depends on the numerical approximation $u_{h,p}$, the local mesh-size h_κ and the local polynomial degree p_κ. To highlight the dependence of the numerical solution on the polynomial degree p, we have explicitly included p as an additional subscript. Then, assuming that $\eta_\kappa(u_{h,p-1}, h_\kappa, p_\kappa - 1) \neq 0$, the perceived smoothness of the solution may be estimated using the ratio

  $$\zeta_\kappa = \eta_\kappa(u_{h,p}, h_\kappa, p_\kappa)/\eta_\kappa(u_{h,p-1}, h_\kappa, p_\kappa - 1),$$

  cf. Adjerid *et al.* [2] and Gui & Babuška [48], for example. If $\zeta_\kappa \leq \gamma$, $0 < \gamma < 1$, the error is decreasing as the polynomial degree is increased, indicating that p–enrichment should be performed. On the other hand, if $\zeta_\kappa > \gamma$ then the element κ is subdivided. Here, γ is referred to as a *type–parameter* [48].

- *Predicted error reduction.* A very closely related technique to the type–parameter strategy is based on refining each element κ in the computational mesh according to the refinement history of κ; cf. [88]. To this end, a predicted (local) error indicator $\eta_\kappa^{\rm pred}$ is computed on the basis of the elemental error indicator η_κ calculated on the previous mesh, together with *a priori* estimates of the expected decay of η_κ after the refinement step has been performed, assuming that the underlying analytical solution is locally smooth. If the error indicator computed on the new mesh is larger than $\eta_\kappa^{\rm pred}$, then κ is subdivided; otherwise p–enrichment is performed, cf., also, [67].

- *'Texas 3–step'.* This strategy was first introduced by J.T. Oden and co-workers [93]; here, the smoothness of the solution to the underlying partial differential equation is not directly taken into account. Step 1 involves initialising various parameters, as well as setting intermediate and final error tolerances $\texttt{TOL}_I$ and $\texttt{TOL}_F$, respectively. Then, keeping the polynomial degree fixed, in Step 2 the mesh is adaptively h–refined in order

to ensure that the error (measured in some appropriate norm) is less than $\texttt{TOL}_I$. In the final third step, the mesh is kept fixed, while the local polynomial degrees are increased to achieve the final error tolerance $\texttt{TOL}_F$. For related work, we refer to the articles [25, 92], and the references cited therein.

- *Mesh optimisation strategy.* In this strategy an optimal refinement is determined for each element in the mesh by directly employing results from approximation theory. More precisely, a *reference solution* $\hat{u}$ is computed on a refined finite element space, where all the elements have been uniformly refined and the polynomial degree p has been globally incremented by one. Then, on each element κ in the original finite element mesh, elemental norms of the projection error between $\hat{u}$ and some suitable finite element projection $\Pi(\hat{u})$ may be computed; here, the error is computed by projecting $\hat{u}$ onto a finite element space employing the original mesh, but with a local polynomial degree $p+1$, as well as on a sequence of finite element spaces corresponding to a local h–refinement of κ that results in the same increase in the number of degrees of freedom as the p–enrichment. The optimal refinement of κ is then chosen to be the one which leads to the smallest projection error; elements in the mesh are then refined based on those that will lead to the greatest decrease in the projection error per degree of freedom. This strategy was first introduced by Rachowicz *et al.* [96]; see also [32, 101] for more recent work.

- *Decay rate of Legendre expansion coefficients.* Mavriplis [87] proposed determining whether the solution is locally smooth or non-smooth by calculating the decay rate of the Legendre expansion coefficients of the solution. More precisely, writing a_i, $i = 0, 1, \ldots$, to denote the ith Legendre coefficient in a one–dimensional expansion of the solution, it is assumed that $a_i \sim C\mathrm{e}^{-\sigma i}$, where C and σ are constants determined by a least–squares best fit. In [87], p–refinement was employed when $\sigma > 1$; otherwise h–refinement was used.

- *Local regularity estimation.* Here, the idea is to directly approximate the local Sobolev regularity index k_κ of the (unknown) analytical solution on each element κ in the computational mesh; then p–refinement is performed on elements where $k_\kappa > p_\kappa + 1$, otherwise h–refinement is employed. This strategy was first proposed by Ainsworth & Senior [4] in the context of norm control for second–order elliptic problems. In [4], the local Sobolev regularity index k_κ was estimated by employing a local error indicator η_κ which was computed by solving a series of local problems with different polynomial degrees; k_κ could then be extracted by employing local *a priori* error bounds for η_κ. Extensions of this method to linear and nonlinear hyperbolic problems were considered in the series of papers [77, 102, 103].

For related work on the design of *a posteriori* error indicators for *hp*–adaptive finite element methods, we refer to [90], and the references cited therein; see also [66] for the development of *hp*–adaptive methods in the context of the Galerkin boundary element method.

Stimulated by the last two strategies, in this section we outline a technique for assessing local smoothness. By monitoring the decay rate of the sequence of coefficients in the Legendre series expansion of a square–integrable function u, we develop a strategy for estimating the size of the Bernstein ellipse of u on a given interval in one–dimension, thereby determining whether u is analytic.

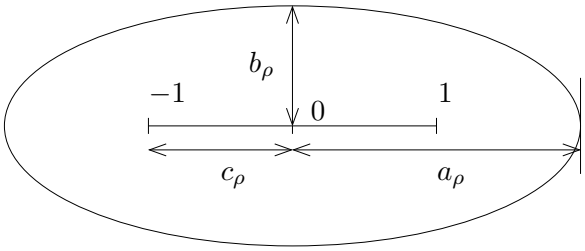


Figure 42: Bernstein ellipse on the interval $[-1, 1]$.

6.4.1 *hp*–extension control

In this section we are concerned with determining whether a given scalar function u is locally analytic on an interval $I_j = (x_{j-1}, x_j)$; here, I_j may be thought of as a given element in a one–dimensional finite element mesh. To address this question, we observe the fact that Legendre coefficients of analytic functions decay to zero at an exponential rate. To describe this precisely, we associate to a function v, defined on the reference domain $\hat{I} = (-1, 1)$, its Bernstein ellipse $\hat{\mathcal{E}}_\rho$ with foci $x = \pm 1$ and radius $\rho = (a_\rho + b_\rho)/c_\rho \geq 1$, where a_ρ and b_ρ are the lengths of the semi–major and semi–minor axes, respectively, and c_ρ is equal to half the length of the interval $\hat{I}$, i.e., $c_\rho = 1$, cf. Figure 42. We remark that $\rho = 1$ corresponds to the degenerate case of $a_\rho = 1$, $b_\rho = 0$ and $\hat{\mathcal{E}}_\rho = [-1, 1]$; thereby, v is singular in $\hat{I}$. With this notation, we have the following result.

Theorem 6.19 *Let $z \mapsto v(z)$ be analytic in the interior of $\hat{\mathcal{E}}_\rho$, $\rho > 1$, but not in the interior of any $\hat{\mathcal{E}}_{\rho'}$ with $\rho' > \rho$. Then the Legendre series*

$$v(z) = \sum_{i=0}^{\infty} b_i L_i(z)\,, \quad b_i = \frac{2i+1}{2} \int_{-1}^{1} v(z) L_i(z)\,\mathrm{d}z \tag{187}$$

converges absolutely and uniformly on any closed set in the interior of $\hat{\mathcal{E}}_\rho$ and diverges in the exterior to $\hat{\mathcal{E}}_\rho$. Moreover,

$$\frac{1}{\rho} = \limsup_{i \to \infty} |b_i|^{1/i}\,. \tag{188}$$

Conversely, if $(b_i)_{i \geq 0}$ is a sequence satisfying (188) with some $\rho > 1$, then the Legendre series (187) converges absolutely and uniformly on any closed set inside of $\hat{\mathcal{E}}_\rho$ to an analytic function $z \mapsto v(z)$ satisfying (187)–(188). The series diverges in the exterior of $\hat{\mathcal{E}}_\rho$.

Proof: See Davis [31], Theorem 12.4.7, for details. □

This result can be localised to the interval $I_j = (x_{j-1}, x_j)$. To this end, we need the family $\left\{ L_i^{[j]}(x) \right\}_{i=0}^{\infty}$ of $L_2(I_j)$–orthogonal polynomials. Using the orthogonality properties of the Legendre polynomials, we find that

$$L_i^{[j]}(x) = (1/h_j)^{1/2}\, L_i\left((x - m_j)/h_j\right)\,,$$

where $h_j = (x_j - x_{j-1})/2$ and $m_j = (x_{j-1} + x_j)/2$. By the completeness of $\left\{L_i^{[j]}(x)\right\}_{i=0}^{\infty}$ in $L_2(I_j)$, we may write

$$u(x)|_{I_j} = \sum_{i=0}^{\infty} a_i^{[j]} L_i^{[j]}(x)\,, \quad \text{where} \quad a_i^{[j]} = \frac{2i+1}{2} \int_{I_j} u(x) L_i^{[j]}(x)\,\mathrm{d}x\,. \tag{189}$$

With this notation, the analogue of Theorem 6.19 on the interval I_j holds verbatim; in this case the elemental Bernstein ellipse $\hat{\mathcal{E}}_{\rho_j}$ has foci at x_{j-1}, x_j and radius $\rho_j = (a_j + b_j)/h_j$, where $a_j \geq h_j$ and b_j are the lengths of the semi-major and semi-minor axes, respectively. Moreover, with the elemental Legendre coefficients of u being defined as in (189), if u is analytic in the interior of $\hat{\mathcal{E}}_{\rho_j}$, but not in the interior of any $\hat{\mathcal{E}}_{\rho_j'}$ with $\rho_j' > \rho_j$, the elemental Bernstein radius satisfies

$$\frac{1}{\rho_j} = \limsup_{i\to\infty} \left|a_i^{[j]}\right|^{1/i} \tag{190}$$

with some $\rho_j > 1$. This result suggests that

$$\theta_j = \frac{1}{\rho_j}, \tag{191}$$

is a measure of size of the domain of analyticity of u *relative to the interval* I_j. Thereby, we deduce that $0 \leq \theta_j \leq 1$; $\theta_j = 0$ corresponds to an entire analytic function, whereas $\theta_j = 1$ corresponds to functions with singular support in $\overline{I}_j$, cf. [68, p. 42].

In practice, we must compute an approximation of θ_j (or, equivalently, of ρ_j) based on the available local Legendre coefficients $a_i^{[j]}$, $i = 0, 1, \ldots, p_j$, of u in I_j. Indeed, motivated by (190), one possible approach would be to approximate θ_j by $\hat{\theta}_j = |a_{p_j}^{[j]}|^{1/p_j}$. This definition may not provide a suitably accurate approximation to θ_j, particularly for functions whose Legendre series expansion have repeating patterns of zero coefficients (occurring, for example, for functions which are locally symmetric or antisymmetric about the midpoint of I_j or for functions which have lacunary series expansions), since only the highest computed Legendre coefficient is included into the criterion.

Thereby, we consider an alternative approach which takes into account *all* of the computed Legendre coefficients on I_j. To this end, employing (190) we deduce that if u is analytic in $\overline{I}_j$ and all subsequences of the sequence $\left\{|a_{p_j}^{[j]}|^{1/p_j}\right\}$ converge to the same limit $1/\rho_j$, then $|a_i^{[j]}| \sim (1/\rho_j)^i$, as $i \to \infty$. This implies that $\log|a_i^{[j]}| \sim i\log(1/\rho_j)$, as $i \to \infty$. We compute an approximate value for θ_j by fitting the slope m_j in $|\log|a_i^{[j]}|| = im_j + b_j$ by linear regression to the already computed $\log|a_i^{[j]}|$ for $i = 0, 1, \ldots, p_j$ (note that for $p_j \geq 1$ there are at least two Legendre coefficients of u per element available). Indeed, the slope m_j of the regression line of the data $\{i, y_i = |\log|a_i^{[j]}||\}_{i=0}^{p_j}$ is computed by

$$m_j = 6\frac{2\sum_{i=0}^{p_j} i y_i - p_j \sum_{i=0}^{p_j} y_i}{(p_j+1)\left((p_j+1)^2 - 1\right)}\,;$$

thereby, the following approximation $\hat{\theta}_j$ to θ_j may be determined:

$$\hat{\theta}_j = e^{-m_j}\,. \tag{192}$$

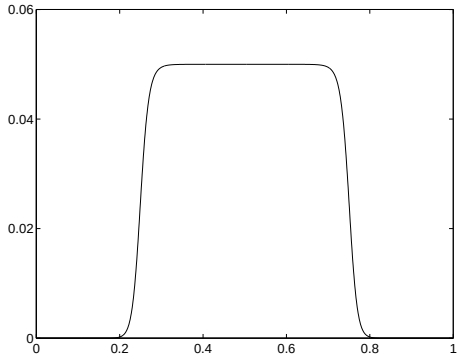


Figure 43: Mixed hyperbolic–elliptic problem. Profile of ε along $y = 0.5$, $0 \leq x \leq 1$.

With this approximation, we can computationally determine the smoothness of a function. The extension of this analyticity estimation procedure to higher–dimensions is based on the application of these techniques in each coordinate direction on a reference element, assuming that a quadrilateral/hexahedral finite element mesh has been employed. For the case of triangular and tetrahedral meshes, we refer to [35]

6.5 Numerical Experiments

In this section we present some numerical experiments to highlight the performance of a goal–oriented *hp*–refinement algorithm, based on estimating the local smoothness of the primal and adjoint solutions by assessing their local analyticity using the algorithm outlined above.

6.5.1 Mixed hyperbolic–elliptic problem

In this first example we investigate the performance of the *hp*–adaptive strategy outlined in Section 6.4 for the interior penalty DG method applied to a mixed hyperbolic–elliptic problem with discontinuous boundary data. We let $a = \varepsilon(\mathbf{x})I$, where

$$\varepsilon = \frac{\delta}{2}(1 - \tanh((r - 1/4)(r + 1/4)/\gamma))\,,$$

$r^2 = (x - 1/2)^2 + (y - 1/2)^2$ and $\delta \geq 0$ and $\gamma > 0$ are constants. Suppose, furthermore, that $\mathbf{b} = (2y^2 - 4x + 1, 1 + y)$, $c = -\nabla \cdot \mathbf{b}$ and $f = 0$.

The characteristics associated with the hyperbolic part of the operator enter the computational domain Ω from three sides of Γ, namely through the vertical edges placed along $x = 0$ and $x = 1$ and the horizontal edge along $y = 0$; the characteristics exit Ω through the horizontal edge along $y = 1$. Thus, on the inflow part of Γ we prescribe the following boundary condition:

$$u(x, y) = \begin{cases} 1 & \text{for } x = 0\ ,\ 0 < y \leq 1\,, \\ \sin^2(\pi x) & \text{for } 0 \leq x \leq 1\ ,\ y = 0\,, \\ \mathrm{e}^{-50y^4} & \text{for } x = 1\,,\ 0 < y \leq 1\,. \end{cases}$$

This is a variant of the test problem presented in [78]. We note that, with $\delta > 0$ and $0 < \gamma \ll 1$, the diffusion parameter ε will be approximately equal to δ in the circular region

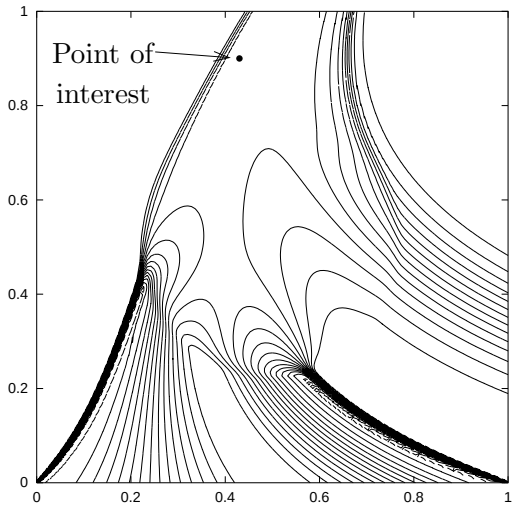


Figure 44: Mixed hyperbolic–elliptic problem. DG approximation to the primal problem on a 129×129 mesh with piecewise bilinear elements ($\mathbf{p}_{\text{iso}} = \mathbf{1}$).

defined by $r < 1/4$, where the underlying partial differential equation is uniformly elliptic. In this example, we set $\delta = 0.05$ and $\gamma = 0.01$; a cross section of ε along $0 \leq x \leq 1$, $y = 1/2$ is shown in Figure 43. As r is increased beyond $1/4$, ε rapidly decreases through a layer of width $\mathcal{O}(\gamma)$; for example, when $r > 0.336$ we have $\varepsilon < 10^{-15}$, so from the computational point of view ε is zero to within rounding error; in this region, the partial differential equation undergoes a change of type becoming, in effect, hyperbolic. Thus we shall refer to the part of Ω with $r > 1/4 + \mathcal{O}(\gamma)$ as the *hyperbolic region*, while the set of points in Ω with $r \leq 1/4$ will be called the *elliptic region*. [Of course, strictly speaking, the partial differential equation is elliptic in the whole of $\bar{\Omega}$.] Furthermore, Figure 44 depicts the numerical approximation to (157)–(159) using the interior penalty DG method on a uniform 129×129 uniform square mesh with $\mathbf{p}_{\text{iso}} = \mathbf{1}$.

Here, we suppose that the aim of the computation is to calculate the value of the analytical solution u at the point of interest $x = (0.43, 0.9)$, i.e.,

$$J(u) = u(0.43, 0.9);$$

cf. Figure 44. The true value of the functional is given by $J(u) = 0.704611313375$.

We first study the performance of our adaptive strategy with h–refinement only, and $\mathbf{p}_{\text{iso}} = \mathbf{1}$. In Table 10 we show the number of nodes, elements and degrees of freedom (DoF) in $V_{h,\mathbf{p}_{\text{iso}}}$, the true error in the functional $|J(u) - J(u_h)|$, the computed *a posteriori* error bound (154) and the corresponding effectivity index θ. Here, we see that the quality of the computed *a posteriori* error bound is extremely good. Indeed, even on relativity coarse meshes, the bound is reliable; moreover, the effectivity index θ shows that $\mathcal{R}_{|\Omega|}(u_h, \bar{z}_h - z_h)$ overestimates the true error in the computed functional by a consistent factor as the finite element space $V_{h,\mathbf{p}_{\text{iso}}}$ is enriched.

In Figure 45 we show the mesh generated after 9 adaptive mesh refinement steps. Here, we see that the mesh is largely concentrated in the neighborhood upstream of the point

# Nodes	# Elements	# Dof	$\lvert J(u) - J(u_h)\rvert$	$\sum_{\kappa\in\mathcal{T}_h} \lvert\bar{\eta}_\kappa\rvert$	θ
81	64	256	7.645e-02	6.597e-02	0.86
119	94	376	2.554e-02	6.331e-02	2.48
206	169	676	9.897e-04	5.640e-02	56.99
357	295	1180	1.323e-03	2.180e-02	16.48
638	538	2152	5.743e-04	8.900e-03	15.50
1053	898	3592	4.959e-04	3.936e-03	7.94
1728	1525	6100	1.453e-04	1.678e-03	11.55
2883	2548	10192	9.295e-05	8.622e-04	9.28
4848	4390	17560	6.002e-05	4.232e-04	7.05
8049	7309	29236	3.323e-05	2.234e-04	6.72
13048	11947	47788	1.562e-05	1.192e-04	7.63

Table 10: Mixed hyperbolic–elliptic problem. Adaptive algorithm using h–refinement

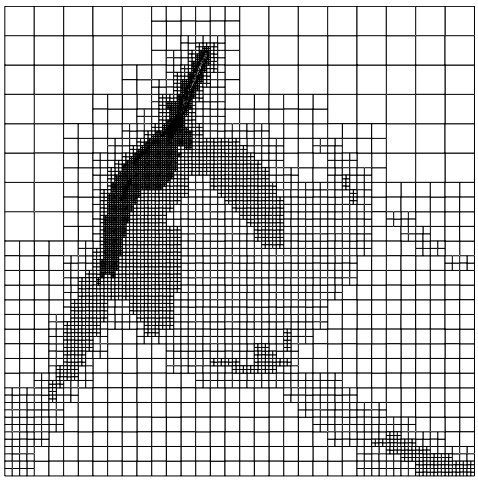


Figure 45: Mixed hyperbolic–elliptic problem. h–mesh after 9 refinements, with 8049 nodes, 7309 elements and 29236 degrees of freedom; here, $|J(u) - J(u_h)| = 3.323 \times 10^{-5}$.

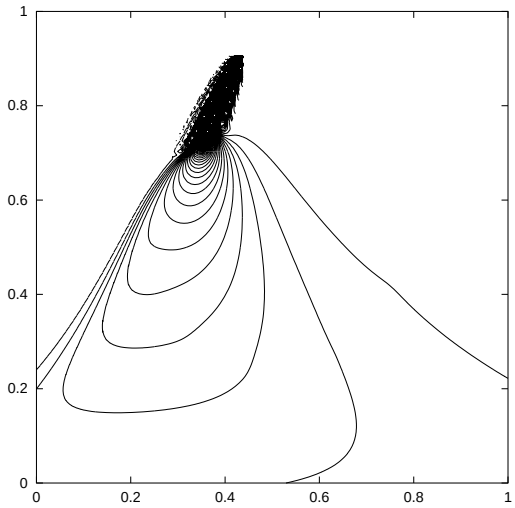


Figure 46: Mixed hyperbolic–elliptic problem. Adjoint solution.

# Nodes	# Elements	# DoF	$\lvert J(u) - J(u_h)\rvert$	$\sum_{\kappa\in\mathcal{T}_h} \lvert\bar{\eta}_\kappa\rvert$	θ
81	64	576	1.924e-02	3.330e-02	1.73
99	76	740	1.056e-02	1.085e-02	1.03
162	130	1451	1.006e-02	2.290e-02	2.28
241	193	2483	7.400e-04	2.385e-03	3.22
302	244	3776	3.760e-05	2.754e-04	7.32
323	262	4777	1.270e-05	1.026e-04	8.08
396	325	6916	9.896e-06	2.245e-05	2.27
487	403	9941	1.224e-06	6.466e-06	5.28
577	481	13528	4.656e-07	1.163e-06	2.50
713	601	19855	2.449e-07	2.582e-07	1.05
960	820	31019	1.574e-08	3.202e-08	2.03
1313	1132	47406	6.531e-10	2.154e-09	3.30

Table 11: Mixed hyperbolic–elliptic problem. Adaptive algorithm using *hp*–refinement

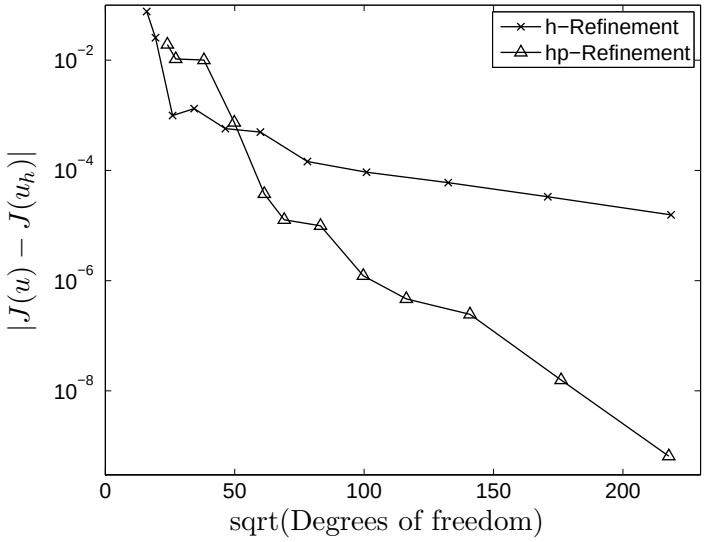


Figure 47: Mixed hyperbolic–elliptic problem. Comparison between h– and hp–adaptive mesh refinement

of interest, together with some almost uniform refinement of the circular region enclosing the part of the computational domain where the underlying partial differential equation is elliptic. We remark that some refinement of the mesh in the region where the discontinuities enter Ω from $(0,0)$ and $(1,0)$, as well as the steep layer entering from the right–hand side boundary has also occurred, though these features of the analytical solution still remain largely unresolved.

The design of the mesh is closely related to the structure of the underlying adjoint solution, since the weighting terms involving the difference between the (approximated) adjoint solution $\bar{z}_h$ and z_h multiply the computable residual terms involving the numerical solution u_h in the definition of the local error indicator $|\bar{\eta}_\kappa|$, cf. (153) with z replaced by $\bar{z}_h$. From Figure 46, we see that in the hyperbolic region of the computational domain above the region of ellipticity, the adjoint solution consists of a single 'spike' originating from the point of interest which is transported upstream along the single characteristic passing through $\mathbf{x} = (0.43, 0.9)$. At the boundary of the circular region where the partial differential equation undergoes a change of type from 'hyperbolic' to elliptic, the spike in the adjoint solution is 'diffused out'. Consequently, the domain of dependence of the point of interest consists of the single characteristic passing through $\mathbf{x} = (0.43, 0.9)$, the circular region where the underlying partial differential equation is elliptic, together with the part of the computational domain enclosed by the intersection of the inflow boundary Γ_- and the two extreme characteristics emanating from the circular elliptic region.

Let us now turn our attention to hp–adaptivity; in Table 11 we show the performance of the proposed adaptive finite element algorithm employing hp–refinement. Here, we again see that the quality of the computed *a posteriori* error bound (154) is extremely good in the sense that it overestimates the true error in the computed functional by a factor of about 1–8. In Figure 47 we plot $|J(u) - J(u_h)|$, using both h– and hp–refinement against the square–root of the number of degrees of freedom on a linear–log scale. We see that after the initial

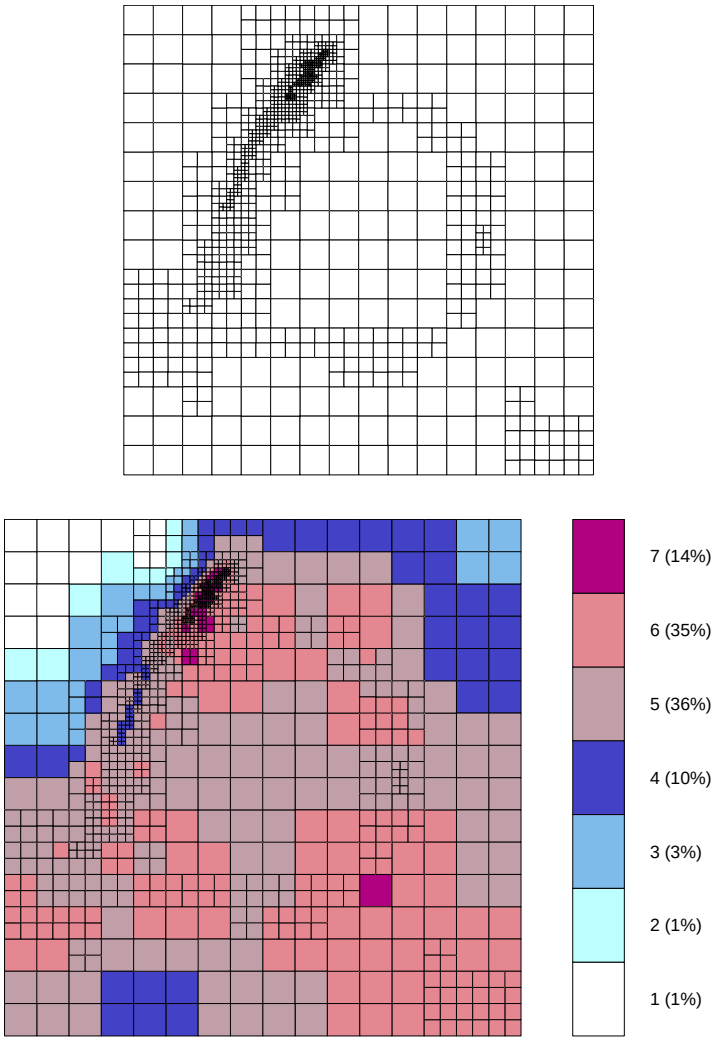


Figure 48: Mixed hyperbolic–elliptic problem. h– and hp–meshes after 11 refinements, with 1313 nodes, 1132 elements and 47406 degrees of freedom; here, $|J(u)-J(u_h)| = 6.531\times10^{-10}$.

# Elements	# Dof	$J_{C_{\rm dp}}(\mathbf{u}) - J_{C_{\rm dp}}(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa$	θ_1	$\sum_{\kappa\in\mathcal{T}_h} \lvert\bar{\eta}_\kappa\rvert$	θ_2
448	7168	-0.4844E-02	-0.4411E-02	0.910	0.4453E-02	0.919
562	10252	-0.1197E-02	-0.1111E-02	0.928	0.1126E-02	0.940
685	14912	-0.5029E-03	-0.4631E-03	0.921	0.4707E-03	0.936
784	19360	-0.3923E-03	-0.3685E-03	0.939	0.3749E-03	0.956
838	23928	-0.1541E-03	-0.1433E-03	0.930	0.1500E-03	0.973
970	31780	-0.7443E-04	-0.6990E-04	0.939	0.7720E-04	1.04
1018	38132	-0.3061E-04	-0.2893E-04	0.945	0.3295E-04	1.08
1045	45616	-0.3010E-04	-0.2770E-04	0.921	0.3009E-04	1.00
1120	56684	-0.7940E-05	-0.7772E-05	0.979	0.9242E-05	1.16
1201	73200	-0.2481E-05	-0.2341E-05	0.944	0.3868E-05	1.56

Table 12: ADIGMA MTC1 test case: *hp*–Refinement algorithm based on an initial structured quadrilateral mesh.

transient, the error in the computed functional using *hp*–refinement becomes (on average) a straight line, thereby indicating exponential convergence of $J(u_h)$ to $J(u)$; this occurs since z is a real analytic function in the regions of the computational domain where u is not smooth and vice verse. Figure 47 also demonstrates the superiority of the adaptive *hp*–refinement strategy over the standard adaptive *h*–refinement algorithm. On the final mesh the true error between $J(u)$ and $J(u_h)$ using *hp*–refinement is over 4 orders of magnitude smaller than the corresponding quantity when *h*–refinement is employed alone.

Figure 48 depicts the primal mesh after 11 adaptive mesh refinement steps. For clarity, we show the *h*–mesh alone, as well as the corresponding distribution of the polynomial degree on this mesh and the percentage of elements with that degree. We see that some *h*–refinement of the primal mesh has occurred in the region of the computational domain upstream of the point of interest, as well as in the circular region where the underlying partial differential equation changes type. Once the *h*–mesh has adequately captured the structure of the primal and adjoint solutions, the *hp*–adaptive algorithm performs *p*–refinement elsewhere in the domain of dependence of the point of interest.

6.5.2 ADIGMA MTC1: Inviscid flow around a NACA0012 airfoil

In this section we consider the performance of the goal–oriented *hp*–refinement algorithm outlined above for the ADIGMA MTC1 test case: inviscid compressible flow around a NACA0012 airfoil with inflow Mach number equal to 0.5, at an angle of attack $\alpha = 2^\circ$. Here, we suppose that the aim of the computation is to calculate the pressure induced drag coefficient $C_{\rm dp}$; i.e., $J(\cdot) \equiv J_{C_{\rm dp}}(\cdot)$. In Tables 12 & 13 we show the performance of the proposed adaptive finite element algorithm employing *hp*–refinement based on exploiting a structured and unstructured (hybrid) starting mesh, respectively. In each case, we show the number of elements and degrees of freedom (Dof) in $V_{h,\mathbf{p}_{\rm iso}}$, the true error in the functional $J_{C_{\rm dp}}(\mathbf{u}) - J_{C_{\rm dp}}(\mathbf{u}_h)$, the computed error representation formula $\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa$, the approximate *a posteriori* error bound $\sum_{\kappa\in\mathcal{T}_h} |\bar{\eta}_\kappa|$, and their respective effectivity indices $\theta_1 = \sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa / (J_{C_{\rm dp}}(\mathbf{u}) - J_{C_{\rm dp}}(\mathbf{u}_h))$ and $\theta_2 = \sum_{\kappa\in\mathcal{T}_h} |\bar{\eta}_\kappa| / |J_{C_{\rm dp}}(\mathbf{u}) - J_{C_{\rm dp}}(\mathbf{u}_h)|$. Here, we see that the quality of the computed error representation formula is extremely good, with $\theta_1 \approx 1$ even on very coarse meshes.

# Elements	# Dof	$J_{C_{\rm dp}}(\mathbf{u}) - J_{C_{\rm dp}}(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa$	θ_1	$\sum_{\kappa\in\mathcal{T}_h} \lvert\bar{\eta}_\kappa\rvert$	θ_2
365	5816	-0.1570E-01	-0.1276E-01	0.813	0.1292E-01	0.823
476	8612	-0.4385E-02	-0.3488E-02	0.795	0.3522E-02	0.803
530	11540	-0.8699E-03	-0.7229E-03	0.831	0.7335E-03	0.843
593	14556	-0.2288E-03	-0.2052E-03	0.897	0.2174E-03	0.950
650	18756	-0.6131E-04	-0.5476E-04	0.893	0.5862E-04	0.956
728	24456	-0.2285E-04	-0.2043E-04	0.894	0.2254E-04	0.986
809	30104	-0.8102E-05	-0.7065E-05	0.872	0.9337E-05	1.15
839	36188	-0.3086E-05	-0.2655E-05	0.860	0.4745E-05	1.54
881	45428	-0.1620E-05	-0.1456E-05	0.899	0.3153E-05	1.95
923	55592	-0.4111E-06	-0.4111E-06	1.00	0.1690E-05	4.11

Table 13: ADIGMA MTC1 test case: *hp*–Refinement algorithm based on an initial unstructured hybrid mesh.

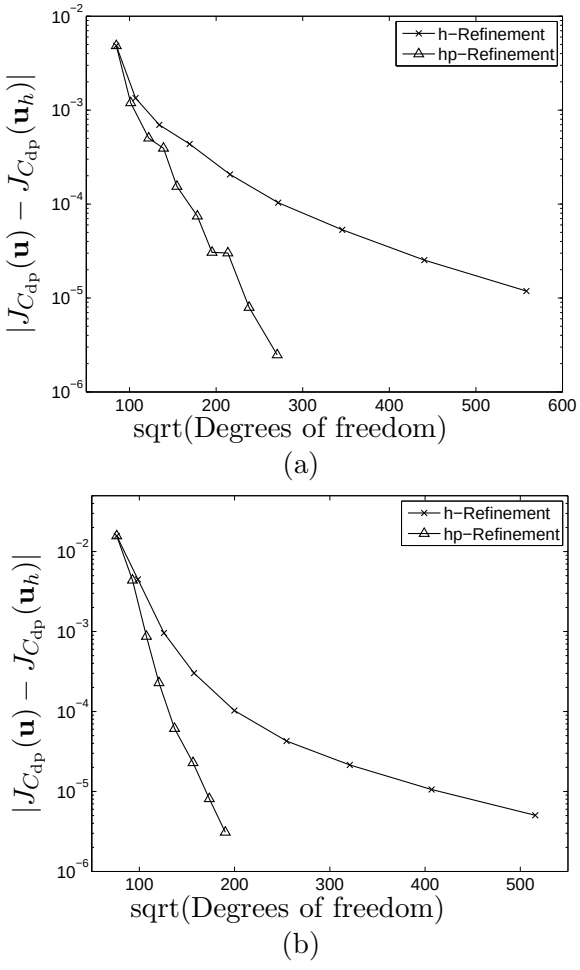
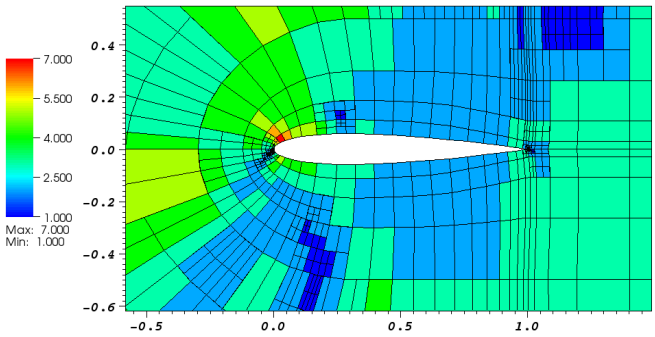
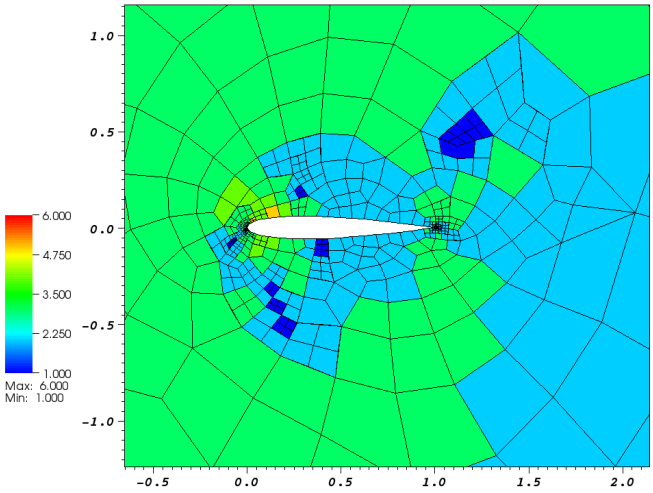


Figure 49: ADIGMA MTC1 test case: Comparison between adaptive *hp*– and *h*–mesh refinement. (a) Structured initial mesh; (b) Unstructured initial mesh.



(a)



(b)

Figure 50: ADIGMA MTC1 test case: *hp*–Mesh distribution. (a) Structured initial mesh after 9 adaptive refinements; (b) Unstructured initial mesh after 7 adaptive refinements.

In Figure 49 we plot the error in the computed target functional $J_{C_{\rm dp}}(\cdot)$, using both *h*– and *hp*–refinement against the square–root of the number of degrees of freedom on a linear–log scale in the case of both a structured and unstructured initial mesh. In both cases, we see that after the initial transient, the error in the computed functional using *hp*–refinement becomes (on average) a straight line, thereby indicating exponential convergence of $J_{C_{\rm dp}}(\mathbf{u}_h)$ to $J_{C_{\rm dp}}(\mathbf{u})$. Figure 49 also demonstrates the superiority of the adaptive *hp*–refinement strategy over the standard adaptive *h*–refinement algorithm. In each case, on the final mesh the true error between $J_{C_{\rm dp}}(\mathbf{u})$ and $J_{C_{\rm dp}}(\mathbf{u}_h)$ using *hp*–refinement is almost 2 orders of magnitude smaller than the corresponding quantity when *h*–refinement is employed alone.

Finally, in Figure 50 we show the *hp*–mesh distributions based on employing a structured and unstructured initial mesh after 9 and 7 adaptive refinement steps, respectively.

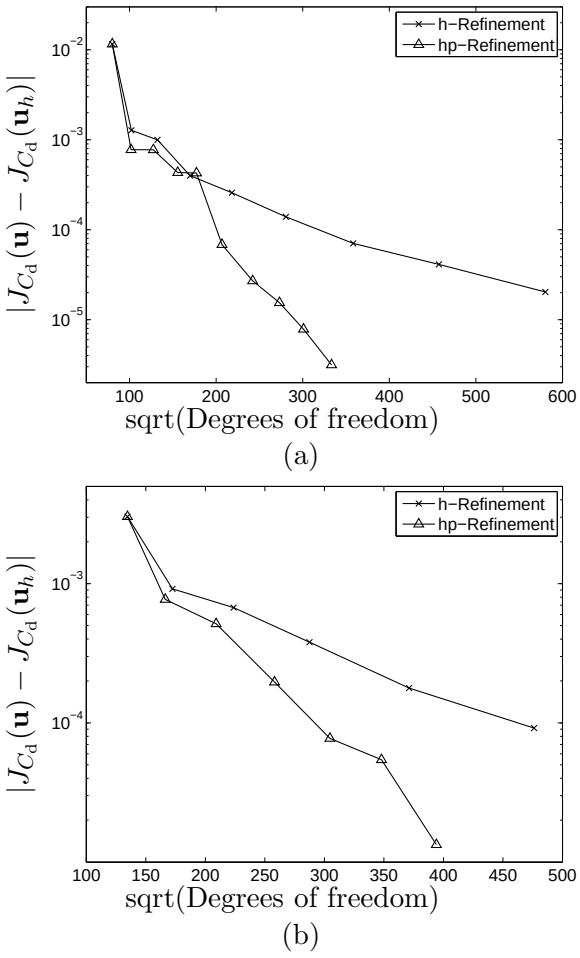
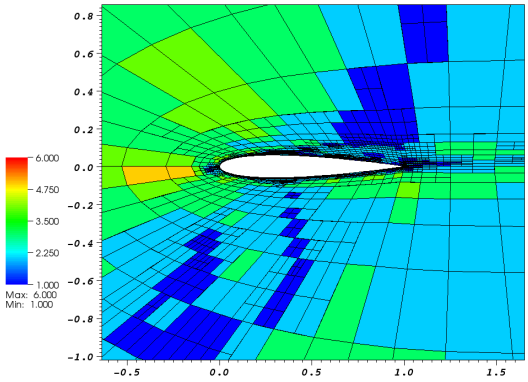


Figure 51: ADIGMA MTC3 test case: Comparison between adaptive *hp*– and *h*–mesh refinement. (a) Structured initial mesh; (b) Unstructured initial mesh.

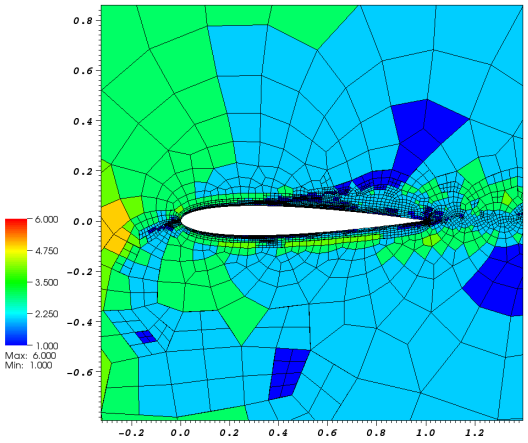
6.5.3 ADIGMA MTC3: Laminar flow around a NACA0012 airfoil

Finally, we consider the ADIGMA MTC3 test case: laminar compressible flow around a NACA0012 airfoil with inflow Mach number equal to 0.5, at an angle of attack $\alpha = 2^\circ$, and Reynolds number $\mathrm{Re} = 5000$ with adiabatic no-slip wall boundary condition imposed on the airfoil geometry. Here, we suppose that the aim of the computation is to calculate the drag coefficient C_d; i.e., $J(\cdot) \equiv J_{C_\mathrm{d}}(\cdot)$.

In Figure 51 we plot the error in the computed target functional $J_{C_\mathrm{d}}(\cdot)$, using both *h*– and *hp*–refinement against the square–root of the number of degrees of freedom on a linear–log scale in the case of both a structured and unstructured initial mesh. As before, in both cases, we see that after the initial transient, the error in the computed functional using *hp*–refinement becomes (on average) a straight line, thereby indicating exponential convergence of $J_{C_\mathrm{d}}(\mathbf{u}_h)$ to $J_{C_\mathrm{d}}(\mathbf{u})$. Figure 51 also demonstrates the superiority of the adaptive *hp*–refinement strategy over the standard adaptive *h*–refinement algorithm. In each case, on the final mesh the true error between $J_{C_\mathrm{d}}(\mathbf{u})$ and $J_{C_\mathrm{d}}(\mathbf{u}_h)$ using *hp*–refinement is over an order of magnitude



(a)



(b)

Figure 52: ADIGMA MTC3 test case: *hp*–Mesh distribution. (a) Structured initial mesh after 8 adaptive refinements; (b) Unstructured initial mesh after 7 adaptive refinements.

smaller than the corresponding quantity when h–refinement is employed alone.

In Figure 52 we show the *hp*–mesh distributions based on employing a structured and unstructured initial mesh after 8 and 7 adaptive refinement steps, respectively. In each case we observe that some h–refinement has been undertaken in the vicinity of the boundary layers as we would expect. However, once the h–mesh has adequately captured the structure of the primal and adjoint solutions, the *hp*–adaptive algorithm subsequently performs p–refinement.

6.6 Anisotropic *hp*–mesh adaptation

In this section, we now consider the general case of automatically generating anisotropically refined computational meshes, together with an anisotropic polynomial degree distribution. With this in mind, once an element has been selected for refinement/derefinement a decision is first be made whether to carry out an h-refinement/derefinement or p-enrichment/derefinement based on the technique developed in Section 6.4, whereby the analyticity of the solutions u and z is assessed by studying the decay rates of their underlying Legendre

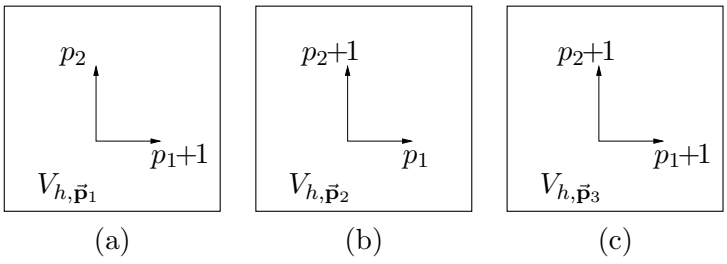


Figure 53: Polynomial Enrichment in 2D: (a) & (b) Anisotropic Enrichment; (c) Isotropic Enrichment.

coefficients. An approximation of the first few Legendre coefficients of u and z are readily obtained from the approximate solutions u_h and $\bar{z}_h$, respectively, and hence a measure of the smoothness of the respective solutions is available for minimal computational effort.

Once the h– and p–refinement flags have been determined on the basis of the above strategy, a decision regarding the type refinement to be undertaken — isotropic or anisotropic — must be made. Motivated by the work in Section 5.7, we employ a competitive refinement technique, whereby the "optimal" refinement is selected from a series of trial refinements. In the h–version setting, we again exploit the algorithm outlined in Section 5.7.

For the case when an element has been selected for polynomial enrichment we consider the p–version counterpart of Algorithm 5.1 and solve local problems based on increasing the polynomial degrees anisotropically in one direction at a time by one degree, or isotropically by one degree. Figure 53 provides a visualisation of the local finite element spaces in two-dimensions, where the original polynomial degree vector on the element of interest is $\mathbf{p} = [p_1, p_2]$. More precisely, we consider the following strategy.

Algorithm 6.1 *This algorithm represents the p–version of Algorithm 5.1 above. Given an element κ in the computational mesh $\mathcal{T}_h$ (which has been marked for p–refinement), we write $V_{h,\vec{\mathbf{p}}}(\kappa)$ to denote the* local *finite element space defined over κ consisting of (continuous) polynomials of composite degree $\vec{\mathbf{p}}$. With this notation, we first construct the local finite element spaces $V_{h,\vec{\mathbf{p}}_i}(\kappa)$, $i = 1, 2, 3$, based on enriching $\vec{\mathbf{p}}$ according to Figures 53(a), (b) and (c), respectively. On each finite element space $V_{h,\vec{\mathbf{p}}_i}(\kappa)$, $i = 1, 2, 3$, we compute the approximate error estimators*

$$\mathcal{R}_{\kappa,i}(u_{h,i}, \bar{z}_{h,i} - z_h) \equiv \bar{\eta}_{\kappa,i},$$

for $i = 1, 2, 3$, respectively. Here, $u_{h,i}$, $i = 1, 2, 3$, is the DG approximation to (157)–(159) computed on the finite element space $V_{h,\vec{\mathbf{p}}_i}(\kappa)$, $i = 1, 2, 3$. Similarly, $\bar{z}_{h,i}$ denotes the DG approximation to z computed on $V_{h,\vec{\mathbf{p}}_i+p_{\text{inc}}}(\kappa)$, $i = 1, 2, 3$, respectively, with polynomials of degree $\vec{\mathbf{p}}_i + p_{\text{inc}}$.

The element κ is then refined according to the subdivision of κ which satisfies

$$\min_{i=1,2,3} \frac{|\eta_\kappa| - |\mathcal{R}_{\kappa,i}(\bar{u}_{h,i}, \bar{z}_{h,i} - z_h)|}{\#dofs(V_{h,\vec{\mathbf{p}}_i}(\kappa)) - \#dofs(V_{h,\vec{\mathbf{p}}}(\kappa))},$$

where $\#dofs(V_{h,\vec{\mathbf{p}}_i}(\kappa))$, $i = 1, 2, 3$, and $\#dofs(V_{h,\vec{\mathbf{p}}}(\kappa))$ denotes the number of degrees of freedom in the local finite element spaces $V_{h,\vec{\mathbf{p}}_i}(\kappa)$, $i = 1, 2, 3$, and $V_{h,\vec{\mathbf{p}}}(\kappa)$, respectively.

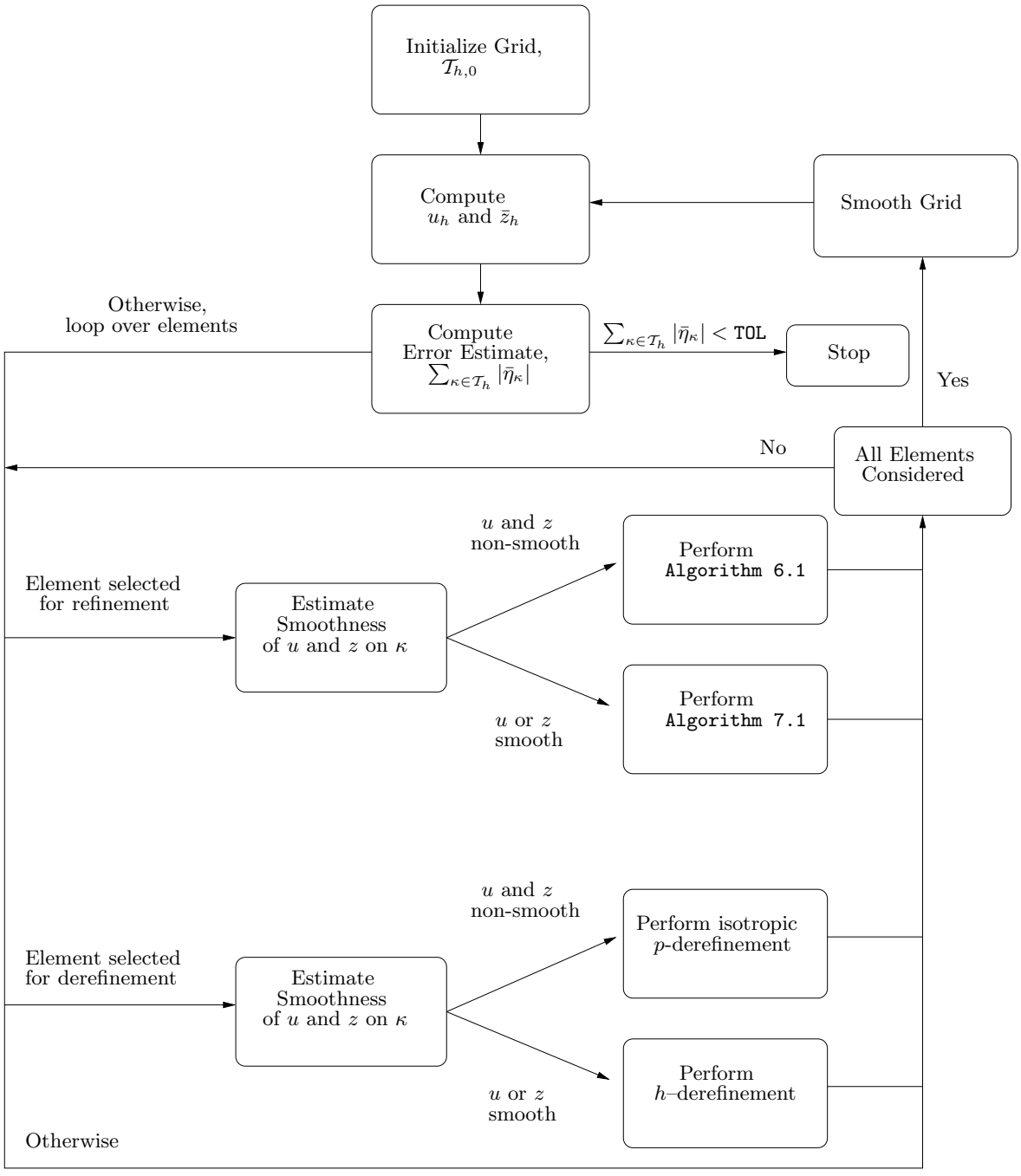


Figure 54: Anisotropic *hp*–adaptive algorithm.

For clarity, the fully anisotropic *hp*-adaptive algorithm presented above can be viewed as a flowchart in Figure 54.

In the following section we shall study the performance of the adaptive anisotropic *hp*–refinement algorithm combining Algorithm 5.1 with Algorithm 6.1.

6.7 Numerical experiments

In this section we present some experiments to assess the numerical performance of the proposed *hp*–anisotropic adaptive algorithm.

6.7.1 Singularly perturbed advection–diffusion problem

We consider the following (singularly perturbed) advection–diffusion problem equation

$$-\varepsilon\Delta u + u_x + u_y = f,$$

for $(x, y) \in (0, 1)^2$, where $0 < \varepsilon \ll 1$ and f is chosen so that

$$u(x, y) = x + y(1 - x) + [\mathrm{e}^{-1/\varepsilon} - \mathrm{e}^{-(1-x)(1-y)/\varepsilon}]\,[1 - \mathrm{e}^{-1/\varepsilon}]^{-1}, \tag{193}$$

cf. Section 5.8.1. Here, we suppose that the aim of the computation is to calculate the value of the (weighted) mean-value of u over the computational domain Ω, i.e.,

$$J(u) = \int_\Omega u\psi \,\mathrm{d}\mathbf{x},$$

where the weight function ψ is chosen as follows:

$$\begin{aligned}\psi(x, y) &= 4(1 - 2y)(1 - \mathrm{e}^{-\alpha(1-x)} - (1 - \mathrm{e}^{-\alpha})(1 - x)) \\ &\quad +4y(y - 1)(\mathrm{e}^{-\alpha(1-x)}(\alpha - (1 - \mathrm{e}^{-\alpha})).\end{aligned}$$

Setting $\alpha = 100$ gives rise to a strong boundary layer in the analytical solution z to the corresponding adjoint problem along the boundary $x = 1$ and a weaker boundary layer along $y = 0$.

Here, we compare the performance of the anisotropic *hp*-refinement adaptive strategy outlined in the previous section with a (standard) isotropic *hp*–refinement strategy, and an *h*–anisotropic/*p*–isotropic refinement algorithm based on employing Algorithm 5.1 to decide the anisotropy in the mesh. In both of these two latter strategies, the decision to perform either *h*– or *p*–refinement/derefinement is again based on estimating the local analyticity of the primal and adjoint solutions u and z, cf. Section 6.4. In all cases, we begin with a uniform (square) mesh with 17 points in each coordinate direction and assign a uniform polynomial degree vector $\vec{\mathbf{p}} = [2, 2]$ on each element.

In Figures 55(a) & (b) we plot the (square root of the) degrees of freedom employed in the finite element space $V_{h,\vec{\mathbf{p}}}$ against the error in the computed target functional $J(\cdot)$, for $\varepsilon = 10^{-2}, 10^{-3}$, respectively, using each of the three *hp*–mesh refinement algorithms defined above. Firstly, we note that in all cases, the convergence lines are (on average) straight, indicating exponential rates of convergence have been achieved using all three refinement strategies for each ε, cf. [44]. Secondly, for each ε, we observe that the computed error, for a given number of degrees of freedom, employing the *h*–isotropic/*p*–isotropic strategy is always inferior to

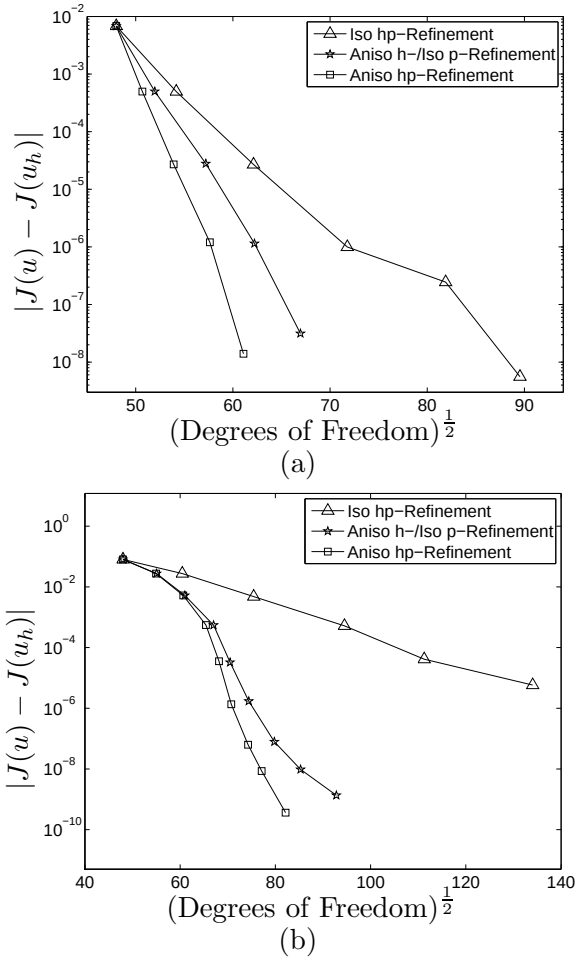
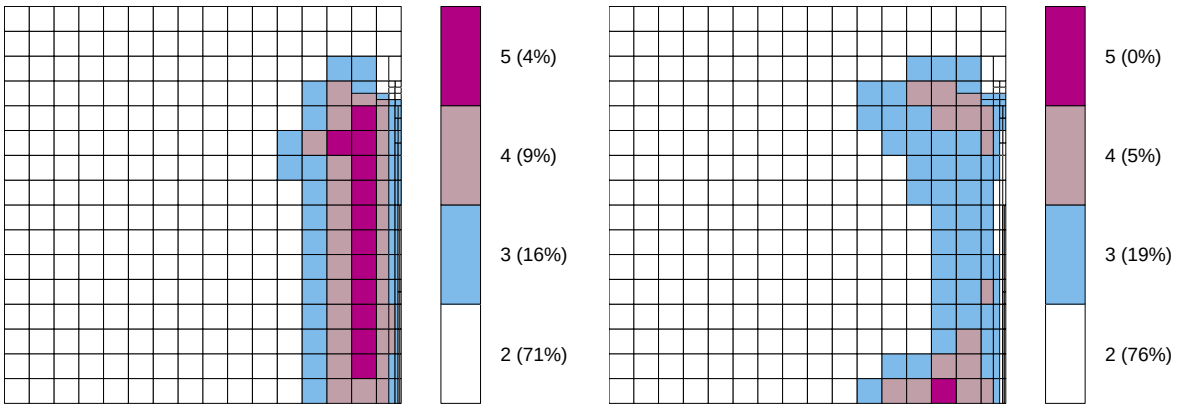


Figure 55: Advection–diffusion problem. Comparison between adaptive *hp*–refinement strategies: (a) $\varepsilon = 10^{-2}$; (b) $\varepsilon = 10^{-3}$.

the algorithm employing *h*–anisotropic/*p*–isotropic refinement. Similarly, this latter strategy is inferior to exploiting the *h*–anisotropic/*p*–anisotropic refinement algorithm outlined in the previous section. Indeed, for $\varepsilon = 10^{-2}$, after the final refinement step, the anisotropic *hp*–strategy yields over two orders of magnitude improvement over the *h*–anisotropic/*p*–isotropic case and nearly 4 orders of magnitude improvement over the isotropic *hp*–method. For $\varepsilon = 10^{-3}$, the anisotropic *hp*–strategy yields around *seven* orders of magnitude improvement in the error in the computed target functional $J(\cdot)$ after the final refinement step, for the same number of degrees of freedom, in comparison to the isotropic *hp*–refinement strategy, and two orders of magnitude improvement over the *h*–anisotropic/*p*–isotropic refinement algorithm. In this latter case, we note that the anisotropic *hp*–refinement algorithm and the *h*–anisotropic/*p*–isotropic strategy perform equally well during the first few refinement steps, since only *h*–adaptation is undertaken. However, as soon as *p*–enrichment is required the use of anisotropic polynomial degrees becomes clearly advantageous. In contrast, in the former case when $\varepsilon = 10^{-2}$, we observe an immediate improvement when employing anisotropic



(a) (b)

Figure 56: Advection–diffusion problem. Anisotropic *hp*-meshes after 4 refinement steps, with 316 elements and 3767 degrees of freedom: (a) p_x and (b) p_y, for $\varepsilon = 10^{-2}$.

hp–adaptivity.

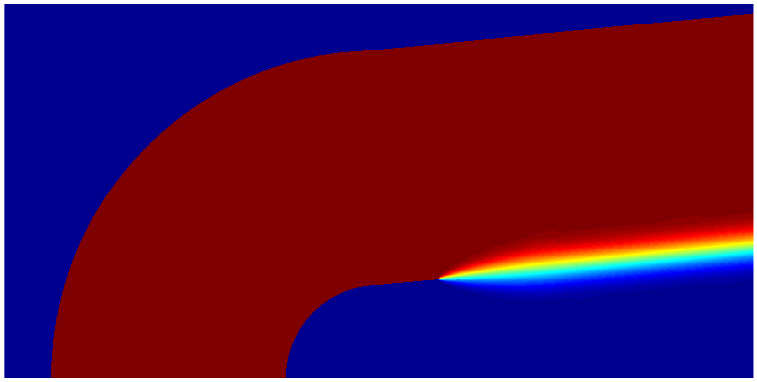
Figure 56 shows the resultant *hp*–mesh distribution after 4 anisotropic *hp*–refinement steps for $\varepsilon = 10^{-2}$; here, Figures 56(a) and (b) show the polynomial degrees employed in the x– and y–directions, respectively. We observe that anisotropic h–refinement has been employed in order to resolve the right–hand side boundary layer and anisotropic p-refinement has been utilized further inside the computational domain. In particular, we notice that the polynomial degrees have been increased to a higher level in the x–direction, than in the orthogonal direction, as we would expect. Quantitatively similar *hp*–mesh distributions are generated for $\varepsilon = 10^{-3}$; for brevity, we omit these results.

6.7.2 Mixed hyperbolic–elliptic problem

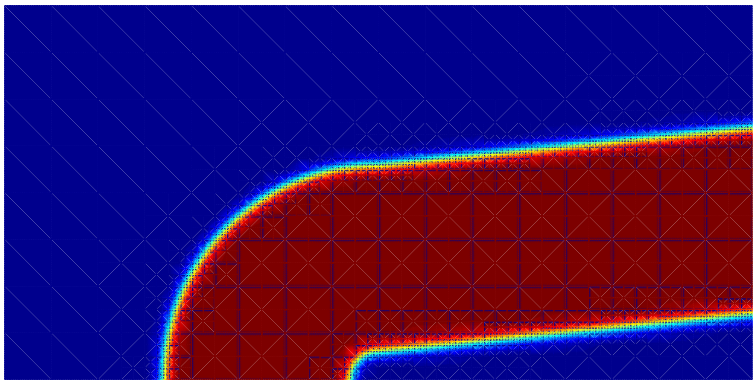
In this second example we investigate the performance of the proposed *hp*–anisotropic refinement algorithm applied to a mixed hyperbolic–elliptic problem with discontinuous boundary data. To this end, we let $\Omega = (0,2) \times (0,1)$, $a = \varepsilon(\mathbf{x})I$, where $\varepsilon = (1 - \tanh(100(r_1 - 0.12)(r_1 + 0.12)))(1 - \tanh(100(r_2 - 0.12)(r_2 + 0.12)))/1000$, $r_1 = x - 1.3$ and $r_2 = y - 0.3$. Furthermore, we set

$$\mathbf{b} = \begin{cases} (y, 1-x)^\top & \text{if } x < 1, \\ (1, 1/10)^\top & \text{if } x \geq 1, \end{cases}$$

$c = 0$, and $f = 0$. On the inflow boundary Γ_-, we select $u(x,y) = 1$ along $y = 0$, $1/8 < x < 3/4$, and $u(x,y) = 0$, elsewhere. This is a variant of the test problem presented in [69]. We note that the diffusion parameter ε will be approximately equal to 3.6×10^{-3} in the square region $(1.18, 1.42) \times (0.18, 0.42)$, where the underlying partial differential equation is uniformly elliptic. As (x,y) moves outside of this region, ε rapidly decreases through a layer of width $\mathcal{O}(0.1)$; for example, when $x = 1.3$ and $y > 0.7$ we have $\varepsilon < 10^{-15}$, so from the computational point of view ε is zero to within rounding error; in this region, the partial differential equation undergoes a change of type becoming, in effect, hyperbolic. Thus, we



(a)



(b)

Figure 57: Mixed hyperbolic–elliptic problem: (a) Primal solution (b) Dual solution.

shall refer to the part of Ω containing this square region (including a strip of size $\mathcal{O}(0.1)$) as the *elliptic region*, while the remainder of the computational domain will be referred to as the *hyperbolic region*. (Strictly speaking, the partial differential equation is elliptic in the whole of $\bar{\Omega}$.) Figure 57(a) shows the analytical solution to the primal problem.

Here, we suppose that the aim of the computation is to calculate the value of the (weighted) outflow advective flux along $x = 2$, $0 \leq y \leq 1$, i.e., $J(u) = \int_0^1 (\mathbf{b} \cdot \mathbf{n}) u(2, y) \psi(y) \, \mathrm{d}y$, where the weight function, in a modification to [43], is

$$\psi(y) = \begin{cases} (\tanh(50(y - 7/40)) + 1)/2 & y < 17/40, \\ (\tanh(-50(y - 27/40)) + 1)/2 & y \geq 17/40; \end{cases}$$

see Figure 57(b) for the corresponding adjoint solution. The true value of the functional is given by $J(u) = 0.324999805677598$.

Once again we compare this anisotropic *hp*–refinement strategy outlined in Section 6.6 with both an *hp*–isotropic algorithm and an *h*–anisotropic/*p*–isotropic refinement strategy, cf. Section 6.7.1. In all cases the starting *hp*–mesh distribution is a uniform 17×9 grid, consisting of uniform square elements, with the uniform polynomial degree distribution $\mathbf{p} = [2, 2]$ on each element.

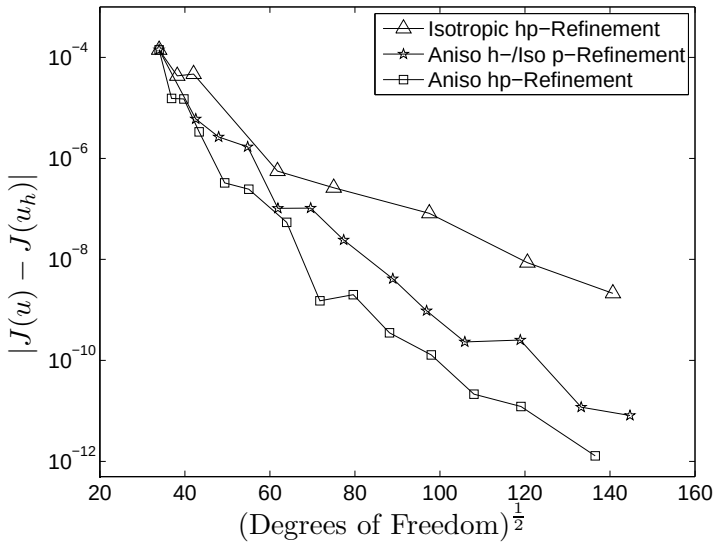


Figure 58: Mixed hyperbolic–elliptic problem: Comparison between adaptive *hp*–refinement strategies.

In Figure 58 we plot the (square root of the) degrees of freedom employed in the finite element space $V_{h,\vec{\mathbf{p}}}$ against the error in the computed target functional $J(\cdot)$, using each of the three *hp*–mesh refinement algorithms defined above, namely *hp*–isotropic refinement, *h*–anisotropic/*p*–isotropic refinement, and *hp*–anisotropic refinement. As in the previous example, we note that in all cases, after an initial transient, the convergence lines are (on average) straight, indicating exponential rates of convergence have been achieved using all three refinement strategies. Similarly, we again observe that the computed error, for a given number of degrees of freedom, employing the *h*–isotropic and *p*–isotropic strategy is always inferior to the algorithm employing *h*–anisotropic and *p*–isotropic refinement, which is in turn inferior to *hp*–anisotropic refinement algorithm. Evidently the majority of improvement over the *hp*–isotropic strategy is due to employing anisotropic *h*–refinement, cf. the previous example when $\varepsilon = 10^{-3}$, yet in the asymptotic regime the *hp*–anisotropic strategy consistently shows around an order of magnitude improvement in the error for the same number of degrees of freedom, when compared with the *h*–anisotropic/*p*–isotropic refinement strategy.

Finally, Figures 59(a) and (b) show the resultant computational mesh and polynomial degree distribution in the x– and y–directions, respectively, after 8 steps of our *hp*–anisotropic refinement strategy. Here, we see that the majority of *h*–refinement has taken place primarily along the layer of the analytical solution u emanating from the point $(x, y) = (3/4, 0)$. In other regions *p*–enrichment has been favoured; indeed there is a marked difference between the polynomial degrees employed in the x– and y–directions, with the majority of elements having had no *p*–enrichment in the x–direction, while most element have had some *p*–enrichment in the y–direction. The *p*–enrichment in the x–direction has been concentrated in the left half of the domain as this is where layers in the primal and adjoint solutions run parallel to the y–axis, while for the same reason *p*–enrichment in the y–direction is concentrated in the right portion of the computational domain.

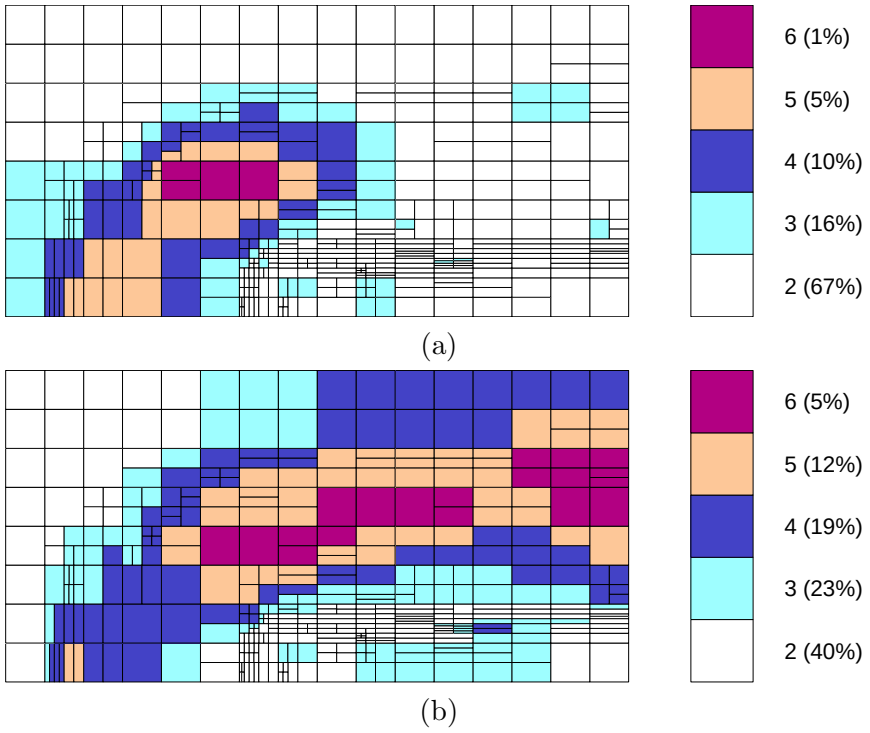


Figure 59: Mixed hyperbolic–elliptic problem. Anisotropic *hp*–meshes after 8 refinement steps, with 410 elements and 6338 degrees of freedom: (a) p_x and (b) p_y.

6.7.3 ADIGMA MTC3: Laminar flow around a NACA0012 airfoil

In this section we again consider the ADIGMA MTC3 test case: laminar compressible flow around a NACA0012 airfoil with inflow Mach number equal to 0.5, at an angle of attack $\alpha = 2^\circ$, and Reynolds number $\mathrm{Re} = 5000$ with adiabatic no-slip wall boundary condition imposed on the airfoil geometry. Here, we suppose that the aim of the computation is to calculate the drag coefficient C_d; i.e., $J(\cdot) \equiv J_{C_\mathrm{d}}(\cdot)$.

In Figure 60 we plot the error in the computed target functional $J_{C_\mathrm{d}}(\cdot)$, using a variety of *h*–/*hp*–adaptive algorithms against the square–root of the number of degrees of freedom on a linear–log scale in the case when an unstructured initial mesh is employed. In particular, here we consider the performance of the following adaptive mesh refinement strategies: isotropic *h*–refinement, anisotropic *h*–refinement, isotropic *hp*-refinement, anisotropic *h*–/isotropic *p*–refinement, and anisotropic *hp*–refinement. Here, we clearly observe that as the flexibility of the underlying adaptive strategy is increased, thereby allowing for greater flexibility in the construction of the finite element space $V_{h,\vec{\mathbf{p}}}$, the error in the computed target functional of interest is improved in the sense that the error in the computed value of $J_{C_\mathrm{d}}(\cdot)$ is decreased for a fixed number of degrees of freedom. However, we point out that in the initial stages of refinement, all of the refinement algorithms perform in a similar manner. Indeed, it is not until the structure of the underlying analytical solution is resolved that we observe the benefits of increasing the complexity of the adaptive refinement strategy. Finally, we point out that the latter three refinement strategies incorporating *p*–refinement all lead to exponential

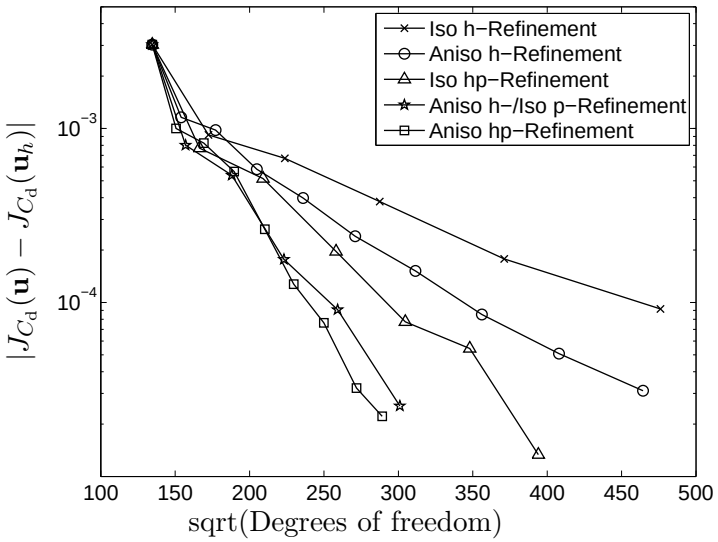


Figure 60: ADIGMA MTC3 test case: Comparison between different adaptive refinement strategies.

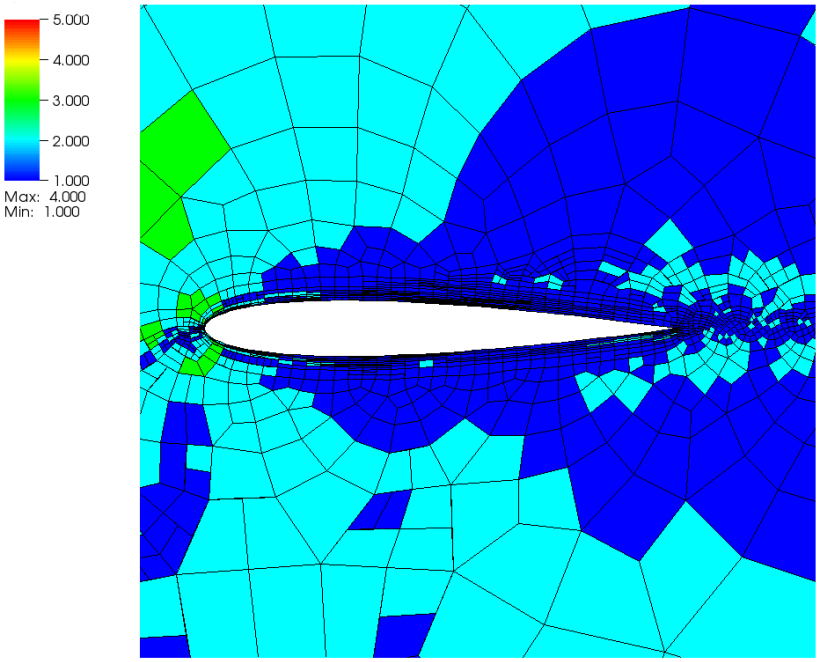


Figure 61: ADIGMA MTC3 test case: h–/p_x–mesh distribution after 5 adaptive anisotropic hp–refinements, with 2200 elements and 52744 degrees of freedom.

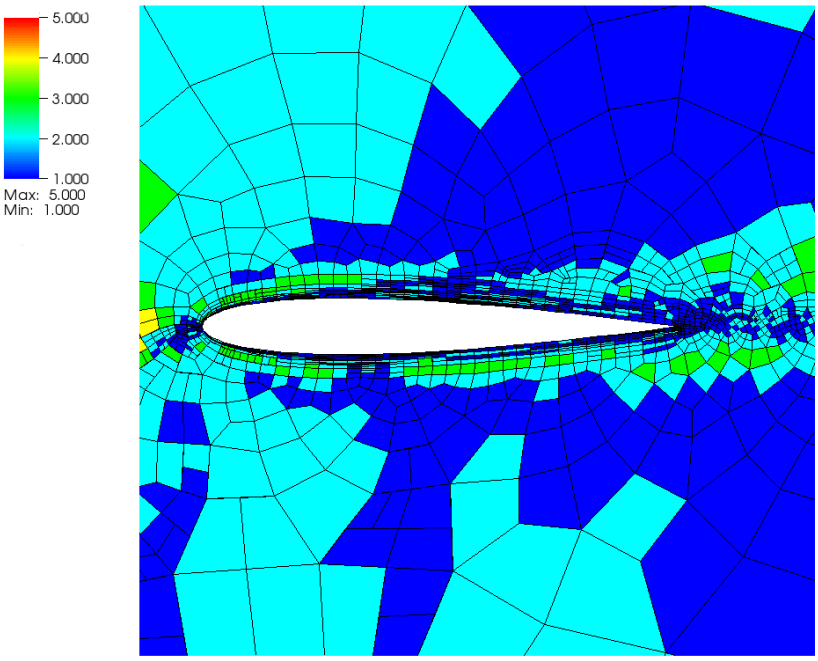


Figure 62: ADIGMA MTC3 test case: h–/p_y–mesh distribution after 5 adaptive anisotropic hp–refinements, with 2200 elements and 52744 degrees of freedom.

convergence of $J_{C_\mathrm{d}}(\mathbf{u}_h)$ to $J_{C_\mathrm{d}}(\mathbf{u})$.

Figures 61 & 62 show the resultant hp–mesh distribution when employing anisotropic hp–refinement after 5 adaptive steps; here, Figures 61 & 62 show the (approximate) polynomial degrees employed in the x– and y–directions, respectively. We observe that anisotropic h–refinement has been employed in order to resolve the boundary layer and anisotropic p-refinement has been utilized further inside the computational domain. In particular, we notice that the polynomial degrees have been increased to a higher level in the orthogonal direction to the curved geometry, as we would expect.

7 Application of error estimation and adaptation to complex flows

In this section the adjoint-based error estimation and mesh refinement algorithms described in Section 4 are applied to complex flows, including three dimensional laminar flows, as well as two and three dimensional turbulent flows. In addition to isotropic refinement, cf. Section 4.5, here we will also use anisotropic mesh refinement. However, here we do not employ the anisotropic mesh refinement algorithm developed in Section 5; instead, we exploit a simpler but still effective anisotropic indicator described the following.

The mesh refinement indicators derived in Sections 4.1 and 4.4 provide only information regarding which elements should be refined in order to improve the accuracy of the resulting solution. As these error indicators do not include any directional information, an additional anisotropic indicator is needed in order to decide whether splitting just a subset of an element's edges and thus modifying the child elements' aspect ratios is preferable over splitting all edges. In the latter case the refinement is isotropic in the sense that child elements inherit the aspect ratio of the mother element. The jump indicator considered here was introduced in [85, 86] for two-dimensional flows. For completeness, we recall the most relevant details and their extension to three-dimensional problems, see [64].

One of the most characteristic features of DG methods is the possible discontinuity of its discrete solutions. In fact, a discrete solution may have jumps across the faces between neighboring elements, whereas it is smooth inside each element. These jumps allow some flexibility in approximating the local properties of the solution. In smooth parts of the solution these jumps tend to zero with successive mesh refinement as the approximate solution is enhanced, i.e., as the error decays. Based on this observation it seems justified to assume that a large jump indicates a larger error as compared to a smaller jump. In view of an anisotropic evaluation a large jump over a face indicates that the mesh size perpendicular to this face is too coarse to sufficiently resolve the solution. In this sense inter–element jumps can be used to derive an anisotropic indicator, that uses information which is specific to the numerical method used to solve the problem. Near discontinuities of the solution, like shocks, the jumps might not tend to zero under mesh refinement. However, in this case a large jump detects this discontinuity and suggests a refinement improving the resolution orthogonal to this feature, which is indeed the correct behavior. Thus, the inter-element jumps can be used as an indicator in both smooth and non-smooth regions of the solution.

In order to obtain directional information, the average jump K_i of a function ϕ over the two opposite faces f_i^j, $j = 1, 2$, perpendicular to one coordinate direction i on the reference element can be evaluated as

$$K_i = \frac{\sum_{j=1}^{2} \left| \int_{f_i^j} [\phi] \, \mathrm{d}s \right|}{\sum_{j=1}^{2} \mathrm{meas}(f_i^j)}, \quad i = 1, 2, 3, \tag{194}$$

where $[\phi] = \phi^+ - \phi^-$ denotes the jump of a scalar function ϕ. Equation (194) provides three distinct values for each element; let K_m denote the maximum value of K_i, $i = 1, 2, 3$. We want to refine along each direction l in which the average jump is not considerably smaller than K_m. In order to quantify *considerably*, we introduce a threshold factor $\theta > 1$. Thus we refine along each direction l for which

$$\theta \, K_l > K_m, \quad l = 1, 2, 3. \tag{195}$$

Depending on the relative sizes of the average jumps in the individual directions, several cases may occur, see Figure 63. If the jump is particularly large in one direction, the element

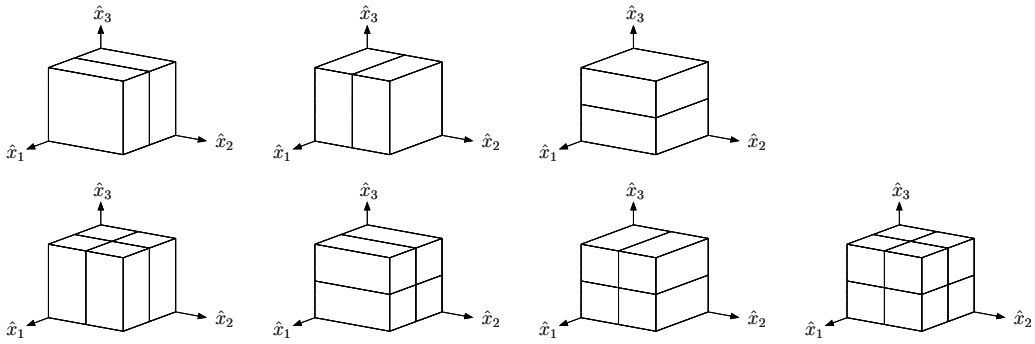


Figure 63: Possible anisotropic and isotropic refinement cases on the 3d reference element.

will be refined only along that direction. If the jump in one direction is particularly small, whereas the other two values are of a similar size, the element will be refined along the other two directions. If all the three average jumps have a similar size, then isotropic refinement will be undertaken.

If the solution function is vector–valued, as is the case for the flow equations, the jump of a scalar function ϕ in Equation (194) has to be replaced by an appropriate norm of the vector of jumps, for example, the L_2-norm.

The empirical threshold factor $\theta > 1$ has to be chosen large enough to ensure that only those elements are flagged for anisotropic refinement, which are located near strong anisotropic features, otherwise the error would not be reduced sufficiently. On the other hand, however, a smaller value of θ allows more elements to be treated anisotropically, thereby leading to a reduced number of total elements. Numerical experiments indicate that $\theta = 5.0$ is a good choice for a range of test problems.

We note that the anisotropic indicators (194) can be used in combination with adjoint-based as well as with residual-based indicators. In contrast to anisotropic indicators which are based on approximation estimates and which include second and possibly higher order derivatives the jump indicators do not rely on the existence of higher order derivatives, see [85, 86] for a more detailed discussion on derivative indicators. For the same reason the jump indicators can easily be extended to the case of *hp*-refined meshes. Finally, the jump indicators are extremely cheap; in particular, they do not require additional local primal and adjoint problems to be solved as required for evaluating error estimates for each of the different refinement cases, see Section 5.

7.1 ADIGMA BTC0: Laminar flow around streamlined body

First, we consider a streamlined three-dimensional body based on a 10 percent thick airfoil with boundaries constructed by a surface of revolution, see Figure 64. It consists of an elliptical leading edge and straight lines. The BTC0 geometry is considered at laminar conditions with inflow Mach number equal to 0.5, at an angle of attack $\alpha = 1^\circ$, and Reynolds number $\mathrm{Re} = 5000$ with adiabatic no-slip wall boundary condition imposed. The geometry

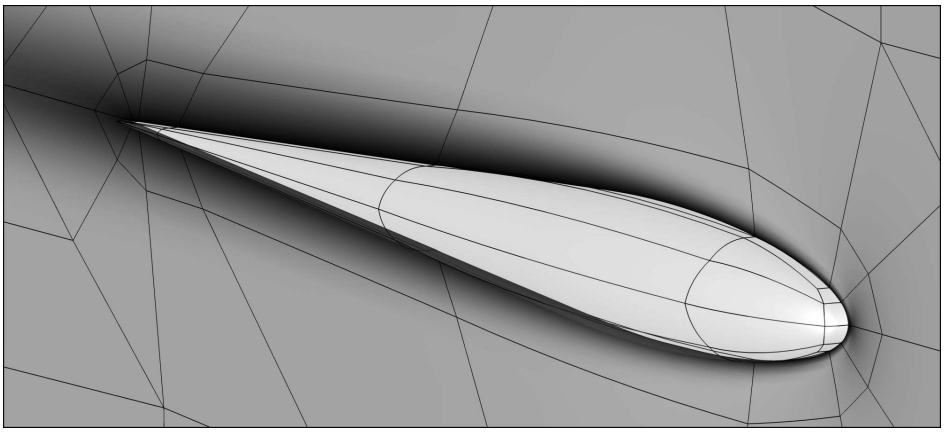


Figure 64: ADIGMA BTC0 test case at laminar conditions: Initial coarse mesh on the body surface and the symmetry plane. The symmetry plane coloring is based on the Mach number distribution computed on a fine mesh, [64].

and the flow is relatively simple. In fact, this test case has been defined in the EU project ADIGMA [82] to enable convergence studies. A reference drag coefficient value of $J_{C_\mathrm{d}}(\mathbf{u}) = 0.063176$ has been obtained by performing high order computations on fine meshes.

We note that in all subsequent computations the boundary of the curved body is approximated using piecewise bi-quadratic polynomials where the additional points required for defining these polynomials are obtained from a CAD representation of the BTC0 geometry. Similarly, also the new points on the boundary required during local mesh refinement near the body are taken from the CAD representation.

The aim of the following computations is to efficiently approximate the drag coefficient on a sequence of locally refined meshes. To this end, we perform the error estimation algorithm described in Section 4.1 on locally refined meshes adapted using the adjoint-based indicators (79) where the adjoint problem (76) is connected to the drag coefficient (51). The first sequence of locally refined meshes is based on isotropic mesh refinement, i.e., upon refinement each hexahedral element is isotropically subdivided into eight hexahedral sub-elements. In Table 14 we collect the number of elements, the number of degrees of freedom (DoF) of $\mathbf{u}_h \in \mathbf{V}_{h,1}$, the true error $J_{C_\mathrm{d}}(\mathbf{u}) - J_{C_\mathrm{d}}(\mathbf{u}_h)$ in the drag coefficient, the estimated error $\mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h) = \sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa$, (78), and the quotient $\theta = \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h) / \left(J_{C_\mathrm{d}}(\mathbf{u}) - J_{C_\mathrm{d}}(\mathbf{u}_h)\right)$ of the estimated and the true error, which is also called the effectivity index. First of all, we see that on all meshes the sign of the error is predicted correctly. On the coarsest three meshes the error estimates are not particularly accurate, which is indicated by an effectivity index θ in the range of $[0.64, 2.7]$. However, as the mesh is refined the effectivity index θ converges to one.

Table 15 collects the corresponding data on a sequence of *an*isotropically refined meshes. Here, on each element depicted for local refinement by the adjoint-based indicators the anisotropic jump indicator (194) is used to determine which of the seven different refinement cases shown in Figure 63 are applied. Here, we see that the error estimation behaves very similar to the one described for the sequence of the isotropically refined meshes in Table 14; in particular, the effectivity of the error estimation does not deteriorate on anisotropically refined meshes.

# Elements	# DoF	$J_{C_\mathrm{d}}(\mathbf{u}) - J_{C_\mathrm{d}}(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa$	θ
768	30720	-9.817e-04	-6.548e-04	0.67
1853	74120	1.737e-03	4.690e-03	2.70
4744	189760	-8.099e-04	-5.146e-04	0.64
12304	492160	-5.007e-04	-4.732e-04	0.95
32282	1291280	-2.825e-04	-2.743e-04	0.97
81688	3267520	-1.063e-04	-1.064e-04	1.00

Table 14: ADIGMA BTC0 test case (laminar): Adaptive algorithm for the accurate approximation of the drag coefficient, C_d, on a sequence of *isotropically* refined meshes, [64].

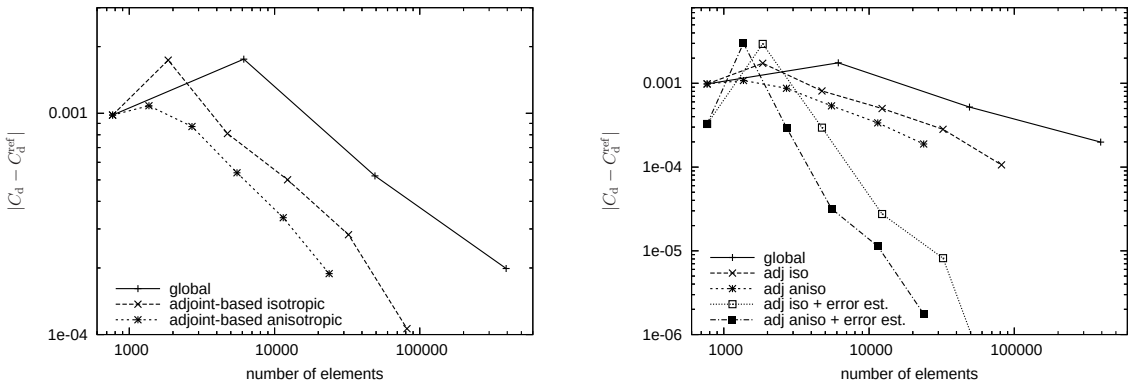
# Elements	# DoF	$J_{C_\mathrm{d}}(\mathbf{u}) - J_{C_\mathrm{d}}(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa$	θ
768	30720	-9.817e-04	-6.548e-04	0.67
1366	54640	1.081e-03	4.096e-03	3.79
2700	108000	-8.711e-04	-5.759e-04	0.66
5518	220720	-5.386e-04	-5.067e-04	0.94
11483	459320	-3.374e-04	-3.261e-04	0.97
23773	950920	-1.886e-04	-1.868e-04	0.99

Table 15: ADIGMA BTC0 test case (laminar): Adaptive algorithm for the accurate approximation of the drag coefficient, C_d, on a sequence of *anisotropically* refined meshes, [64].

Finally, Figure 65(a) plots the error in the drag coefficient $|J_{C_\mathrm{d}}(\mathbf{u}) - J_{C_\mathrm{d}}(\mathbf{u}_h)|$ against the number of elements for a sequence of globally refined meshes, the sequence of adjoint-based isotropic refined meshes, see Table 14, and the sequence of adjoint-based anisotropic refined meshes, see Table 15. Comparing the histories of global and adjoint-based isotropic refinement we see in Figure 65 that for this test case the adjoint-based refinement leads to meshes with a factor of about 5 less elements for a specific accuracy in the drag coefficient when compared to global refinement. Moreover, we see that there is another factor of about 2 in the mesh sizes required for a specific accuracy for the anisotropic algorithm when compared to the isotropic adjoint-based mesh refinement.

A comparison of the resulting meshes for the two refinement algorithms is given in Figure 66. As anisotropic features are not particularly strong and the initial mesh already shows some anisotropy, the overall effect of anisotropic refinement seems rather weak. However, we note that the strong stretching of some cells along the body with small edge length orthogonal to the flow is too pronounced in the initial mesh. During isotropic refinement this aspect ratio is inherited to all child elements. The anisotropic refinement algorithm can modify aspect ratios, however, and it does so in this test case, but in contrast to initial expectations it is reducing aspect ratios in order to find the mesh best fitted to the (quite isotropic) problem at hand, which in this case is a more isotropic mesh. In addition to that, other parts of the adapted mesh show the more common case of elements which have a larger aspect ratio in the anisotropic case.

The error estimates in Tables 14 and 15 can be used to enhance the computed drag coefficients $J_{C_\mathrm{d}}(\mathbf{u}_h)$ as follows: $\tilde{J}_{C_\mathrm{d}}(\mathbf{u}_h) := J_{C_\mathrm{d}}(\mathbf{u}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)$. For smooth solutions such enhanced target quantities can be expected to converge to the true value with higher order of convergence than the original values $J_{C_\mathrm{d}}(\mathbf{u}_h)$. This is confirmed in Figure 65(a)



(a) (b)

Figure 65: ADIGMA BTC0 test case (laminar): Convergence of the error in the drag coefficients $J_{C_\mathrm{d}}(\mathbf{u}_h)$ for global in comparison to adjoint-based isotropic and anisotropic mesh refinement. Additionally, b) shows the errors of the enhanced drag coefficients $\tilde{J}_{C_\mathrm{d}}(\mathbf{u}_h) = J_{C_\mathrm{d}}(\mathbf{u}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)$ for the adjoint-based isotropic and anisotropic mesh refinement, [64].

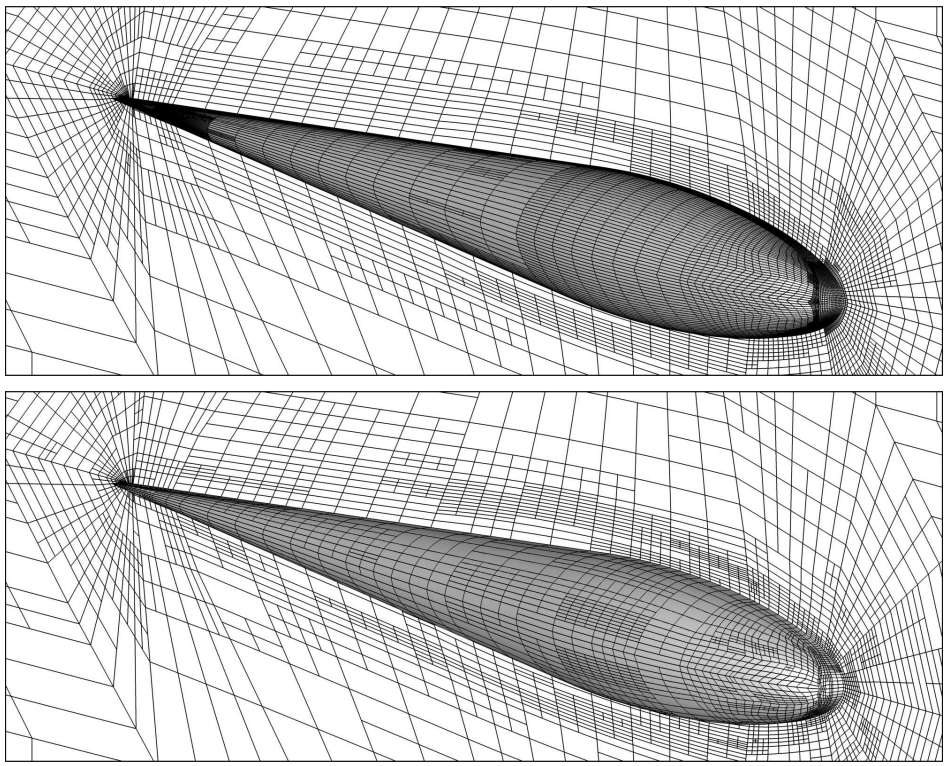


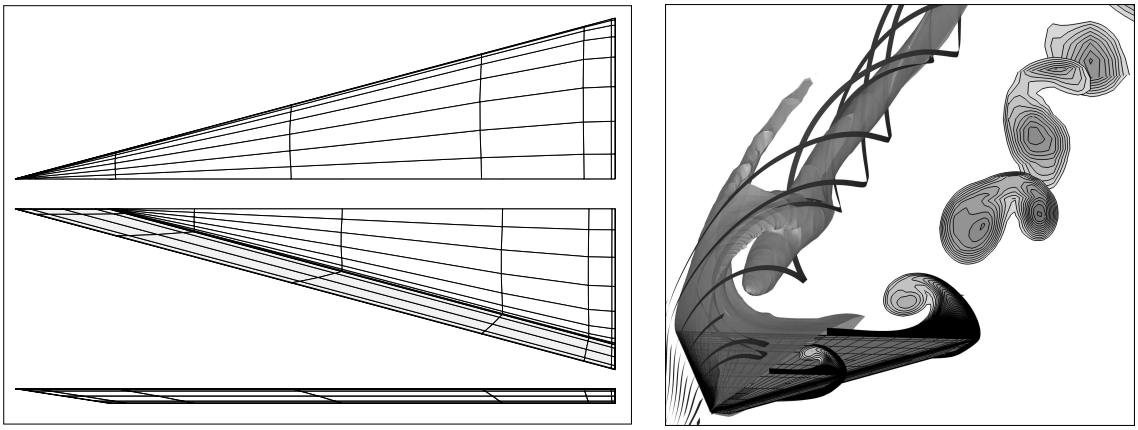
Figure 66: ADIGMA BTC0 test case (laminar): Adapted surface meshes after five adaptation cycles: top: isotropic refinement, bottom: anisotropic refinement, [64].

which repeats the convergence histories of Figure 65(b) in a different scale and additionally shows the histories of the errors of the enhanced drag coefficients $\tilde{J}_{C_\mathrm{d}}(\mathbf{u}_h)$. In fact, from the

third coarsest mesh onwards the enhanced drag coefficients are significantly more accurate than the original values $J_{C_\mathrm{d}}(\mathbf{u}_h)$ and converge with higher order.

7.2 ADIGMA BTC3: Laminar flow around delta wing

As a second test case we consider a laminar flow around a delta wing. The delta wing has a sloped and sharp leading edge and a blunt trailing edge. A similar case has previously been considered in [81]. The geometry of the delta wing can be seen from the initial surface mesh in Figure 67(a). The delta wing is considered at laminar conditions with inflow Mach



(a) (b)

Figure 67: Laminar delta wing: a) initial surface mesh: Top, bottom and side view of the half delta wing with straight leading edges, b) solution plot showing streamlines and a Mach number isosurface over the left half of the wing as well as Mach number slices over the right half, [64].

number equal to 0.3, at an angle of attack $\alpha = 12.5°$, and Reynolds number $\mathrm{Re} = 4000$ with isothermal no-slip wall boundary condition imposed on the wing geometry. This is the ADIGMA BTC3 test case as defined in the EU project ADIGMA [82]. As the flow passes the leading edge it rolls up, creates a vortex and a secondary vortex. The resulting vortex system remains over long distances behind the wing, see Figure 67(b).

By performing high order computations on fine meshes the following reference values of the force coefficients have been obtained: $J_{C_\mathrm{d}}(\mathbf{u}) = 0.16608$ and $J_{C_\mathrm{l}}(\mathbf{u}) = 0.34865$. In the following we will compare the performance of the adjoint-based mesh refinement algorithm for the accurate approximation of the drag and lift coefficients with both a residual-based strategy and global mesh refinement. Additionally, for the local mesh refinement strategies we will compare isotropic against anisotropic mesh refinement.

Let us first consider the lift coefficient; performing the error estimation and adjoint-based mesh refinement algorithm with the adjoint problem connected to the lift coefficient, we collect the data of the sequence of isotropically refined meshes in Table 16. Here we see that already on the coarse meshes the error estimation is quite accurate and it improves as the mesh is refined. A similar behavior is also seen for anisotropic mesh refinement, cf. Table 16, where the error estimation is slightly more accurate than in the isotropic case.

# Elements	# DoF	$J_{C_l}(\mathbf{u}) - J_{C_l}(\mathbf{u}_h)$	$\sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa$	θ
3264	130560	-2.686e-02	-2.022e-02	0.75
8346	333840	-1.639e-02	-1.232e-02	0.75
22647	905880	-9.017e-03	-7.867e-03	0.87
60524	2420960	-4.537e-03	-4.715e-03	1.04

Table 16: Laminar delta wing: Adaptive algorithm for the accurate approximation of the lift coefficient on a sequence of isotropically refined meshes, [64].

# Elements	# DoF	$J_{C_l}(\mathbf{u}) - J_{C_l}(\mathbf{u}_h)$	$\sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa$	θ
3264	130560	-2.686e-02	-2.022e-02	0.75
6347	253880	-1.767e-02	-1.470e-02	0.83
14108	564320	-8.855e-03	-7.405e-03	0.84
32331	1293240	-4.605e-03	-4.612e-03	1.00

Table 17: Laminar delta wing: Adaptive algorithm for the accurate approximation of the lift coefficient on a sequence of anisotropically refined meshes, [64].

Figure 68(a) plots the error in the lift coefficient $|J_{C_l}(\mathbf{u}) - J_{C_l}(\mathbf{u}_h)|$ against the number of elements for various refinement strategies: global mesh refinement, residual-based isotropic and anisotropic mesh refinement, as well as adjoint-based isotropic and anisotropic mesh refinement. We notice that lift coefficients of a specific accuracy are obtained with less elements for residual-based mesh refinement than for global mesh refinement where this advantage increases for increasing accuracy requirements. Furthermore, there is a significant decrease of the number of elements required for a specific accuracy when comparing adjoint-based against residual-based refinement. Additionally, in case of adjoint-based mesh refinement Figure 68(a) plots the errors of the enhanced lift coefficients $\tilde{J}_{C_l}(\mathbf{u}_h) := J_{C_l}(\mathbf{u}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)$. We note that already on the coarsest mesh the enhanced lift coefficient is almost as accurate as the lift coefficients on the finest adjoint-based refined mesh. Finally, we see that anisotropic mesh refinement always performs better than isotropic mesh refinement. In fact, anisotropic adjoint-based refinement requires about half the number of elements for almost the same accuracy than the corresponding isotropic refinement.

Finally, we consider the drag coefficient. Tables 18 and 19 collect the data of the sequences of, respectively, the isotropically and anisotropically adjoint-based refined meshes. In both cases we see a analogous behavior to that described for the lift coefficient. Finally, Figure 68(b) plots the errors for global refinement, residual-based refinement (isotropic and anisotropic), adjoint-based refinement (isotropic and anisotropic) and the errors of the enhanced lift coefficients. Here, we again observe behavior very similar to that described for the lift coefficient above.

Adapted meshes for the six different combinations of error indicators and isotropic or anisotropic refinement are presented in Figure 69.

All plots are given for the last data point in the errors plots in Figure 68, so the accuracy for the relevant target functional values is comparable. The outstanding effect is clearly the resolution of the vortex in the cut-plane behind the wing for the residual-based refinement indicator and the corresponding lack of resolution in this area in the case when goal-oriented refinement is employed. It is quite obvious that the global flow field is better resolved using

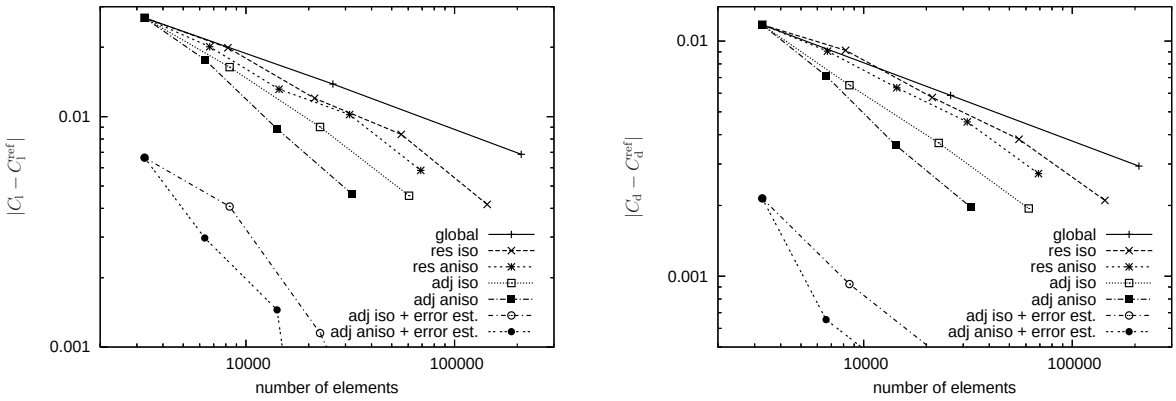


Figure 68: Laminar delta wing: Convergence of the error in the a) lift and b) drag coefficients $J(\mathbf{u}_h)$ for global in comparison to residual-based (isotropic and anisotropic) and to adjoint-based (isotropic and anisotropic) mesh refinement. Additionally, the errors of the enhanced force coefficients $\tilde{J}(\mathbf{u}_h) = J(\mathbf{u}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)$ on the sequences of adjoint-based mesh refinement are given, [64].

# Elements	# DoF	$J_{C_\mathrm{d}}(\mathbf{u}) - J_{C_\mathrm{d}}(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa$	θ
3264	130560	-1.174e-02	-9.594e-03	0.82
8549	341960	-6.492e-03	-5.566e-03	0.86
22885	915400	-3.688e-03	-3.223e-03	0.87
61868	2474720	-1.942e-03	-1.941e-03	1.00

Table 18: Laminar delta wing: Adaptive algorithm for the accurate approximation of the drag coefficient on a sequence of *isotropically* refined meshes, [64].

# Elements	# DoF	$J_{C_\mathrm{d}}(\mathbf{u}) - J_{C_\mathrm{d}}(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa$	θ
3264	130560	-1.174e-02	-9.594e-03	0.82
6600	264000	-7.118e-03	-6.464e-03	0.91
14215	568600	-3.615e-03	-3.230e-03	0.89
32621	1304840	-1.967e-03	-1.928e-03	0.98

Table 19: Laminar delta wing: Adaptive algorithm for the accurate approximation of the drag coefficient on a sequence of *anisotropically* refined meshes, [64].

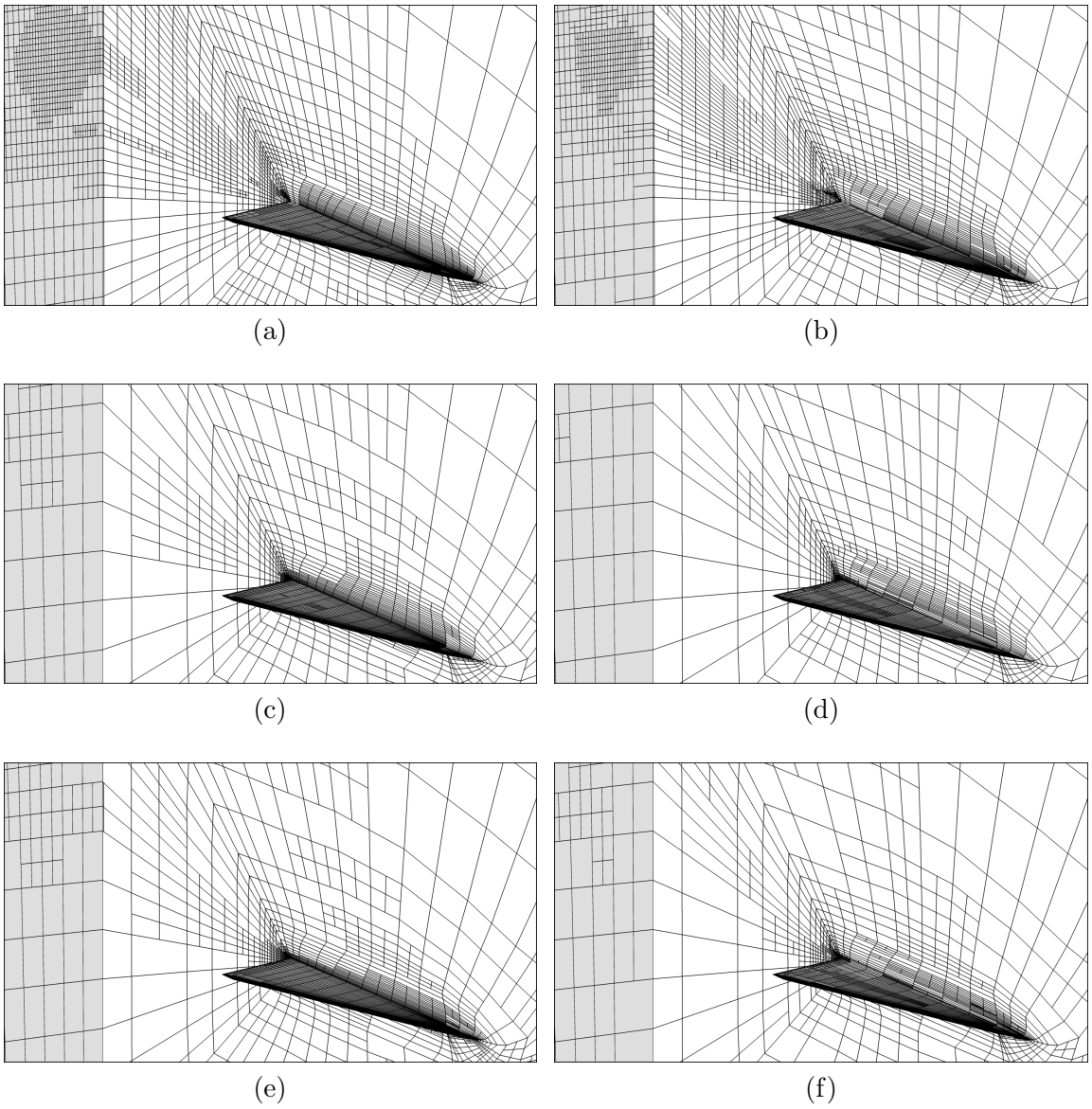
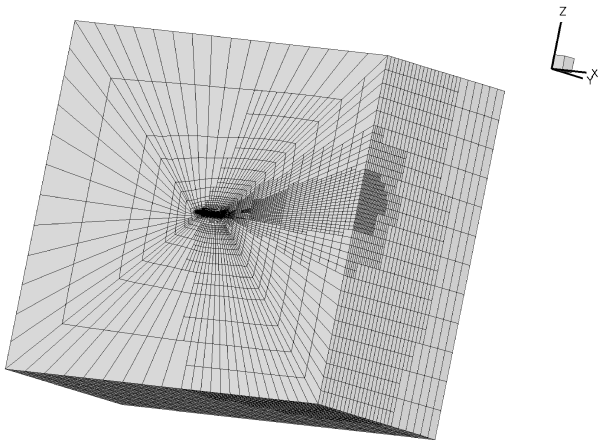
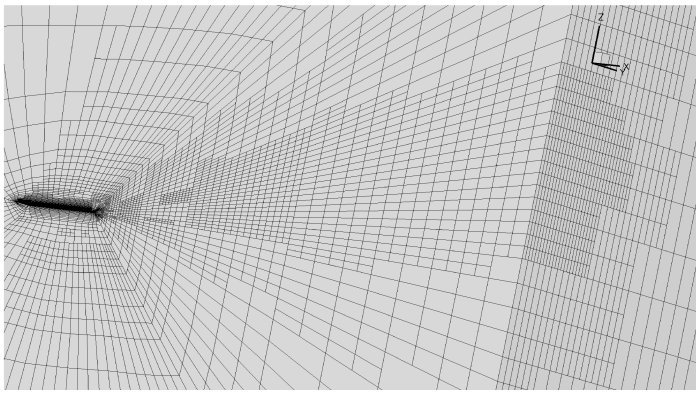


Figure 69: Laminar delta wing: Adapted meshes, a) and b): four adaptation steps with the residual indicator, isotropic and anisotropic, c) and d): three adaptation steps with the adjoint-based indicator for the lift coefficient C_{l}, isotropic and anisotropic, e) and f): three adaptation steps with the adjoint-based indicator for the drag coefficient C_{d}, isotropic and anisotropic, [64].



(a)



(b)

(c)

Figure 70: Laminar delta wing: Mesh after 4 isotropic residual-based refinement steps. (a) Distant view; (b) Close up view; (c) Mach number distribution. The vortex is resolved till the outflow boundary. Result of the PADGE code [57].

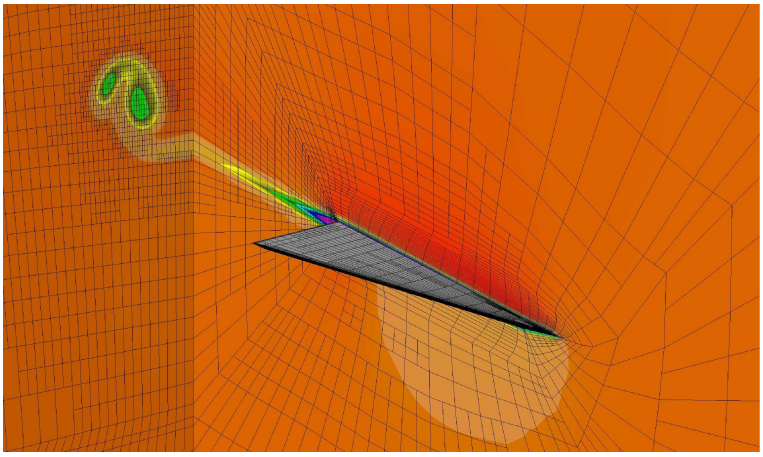


Figure 71: Laminar delta wing: Mach number distribution on mesh after 4 isotropic residual-based refinement steps. Result of the PADGE code [57].

the former type of indicator, whereas the resolution of this prominent vortex does not strongly influence the target functional values, as both the pressure at the wall and the skin friction are only weakly dependent on the downstream vortex evolution. Thus, investing more in the near-wall refinement, the adjoint-based refinement indicators are capable of creating more efficient meshes for the approximation of a given target functional. In contrast to that, the residual-based indicator is particularly well suited for resolving the overall flow field. In Figure 69(a) we see that the vortex system is well resolved in a cut-plane not too far behind the wing. Moreover, the vortex is well resolved over a significantly longer distance. In fact, the distant and close up view of the mesh depicted in Figures 70(a)&(b) shows that refinement along the path of the vortex is undertaken up to the outflow boundary. This way the vortex is kept and resolved until the outflow boundary, see Figure 69(c).

7.3 ADIGMA BTC1: L1T2 high-lift configuration

In this section we consider a turbulent flow around a typical high-lift configuration, the L1T2 three-element airfoil. The geometry of this configuration is outlined in Fig. 73(a). In particular, we consider a turbulent flow at Mach number $M = 0.197$, Reynolds number $Re = 3.52 \cdot 10^6$ and an angle of attack $\alpha = 20.18°$. This case has been documented extensively in the literature, see e. g. [37, 65]. In particular, there is data of two wind tunnel experiments available, see [89]; in the sequel we refer to this data as experiment 1 and experiment 2. Furthermore, this test case has been considered as test case BTC1 in the EU project ADIGMA [82].

In the following computations based on the PADGE code [57] a DG discretization of the RANS-$k\omega$ equations is used which represents a slight modification of the BR2 scheme proposed in [13]. In particular, we use the derived variable $\ln \omega$ instead of ω itself. Furthermore, we impose specific limitations of the turbulence variables. Whereas k is simply kept non-negative, the variable ω is bounded from below by a local, i.e., pointwise minimal ω_0 value derived from the realizability of the Reynolds stresses, see [13]. At wall boundaries we use Menter's boundary condition for ω, where the first wall boundary layer grid spacing y_1

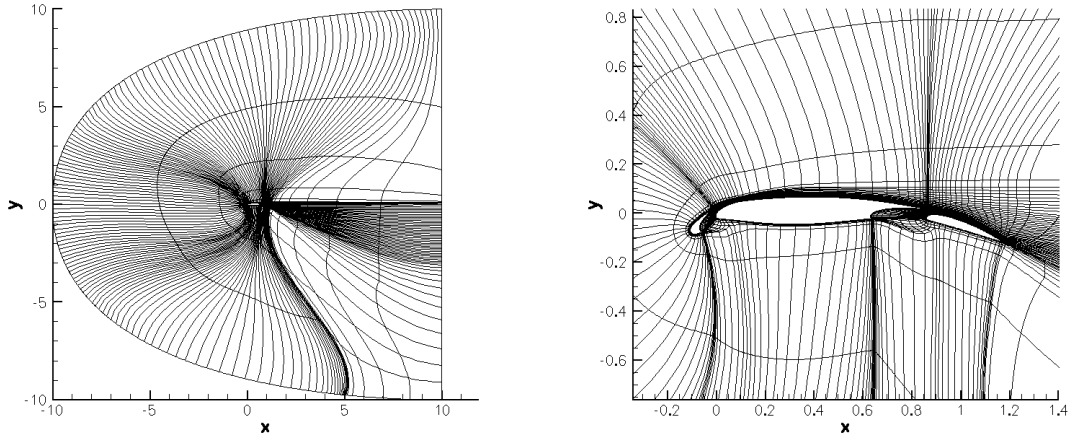


Figure 72: L1T2 high lift configuration: Coarse grid of 4740 curved elements.

is chosen such that y_1^+, i.e., the first $y^+ = y_1 \frac{u_\tau}{\nu}$ value, is in the range of one.

First, we compare numerical results generated by the PADGE code with results generated by the well validated finite volume code TAU [100] as well as with experimental data. The PADGE computations were performed with polynomial degrees $p = 3$ and $p = 4$, each on the same quadrilateral mesh with 4 740 curved elements, see Figure 72. This mesh emerged from an original 75 840 element mesh by two agglomeration steps. The curved mesh representation in this case is realized by piecewise quartic approximation based on extra point data which have been extracted from the original mesh. Reference results have been produced on the original mesh by means of the TAU code.

Figure 73(b) shows the pressure distribution over each of the airfoil elements, i.e., slat, main element and flap. Here, we see that the output by the PADGE code is in a good agreement with the experimental data and with only minor differences compared to the TAU reference results. Furthermore, Figure 73(c) shows the comparison for the skin friction distribution. Whereas there are still considerable differences between the computed skin friction distribution for $p = 3$, the result for $p = 4$ is overall in a good agreement with the TAU reference computation. We note that the $p = 4$ computation took nearly the same number of degrees of freedom as the TAU code.

In the following, we investigate the performance of the adjoint-based and the residual-based mesh refinement for this test case. Starting with $p = 1$ solution on the coarse mesh of 4 740 curved elements, we first consider the adjoint-based refinement targeted at efficiently approximating the lift coefficient C_l. In Figure 74(a) we compare the convergence of C_l for the global, the residual-based and the adjoint-based mesh refinement. We see that with the adjoint-based refinement the C_l value converges significantly faster to the C_l reference value than when residual-based or global mesh refinement are employed. Furthermore, we see that using the error estimation on the adjoint-based refined meshes for computing enhanced lift values $\tilde{J}_{C_\mathrm{l}}(\mathbf{u}_h) = J_{C_\mathrm{l}}(\mathbf{u}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)$ further improves the C_l value. In Figure 74(b) we see the respective plot for the C_d value. Figure 76 and Figure 77 show the final

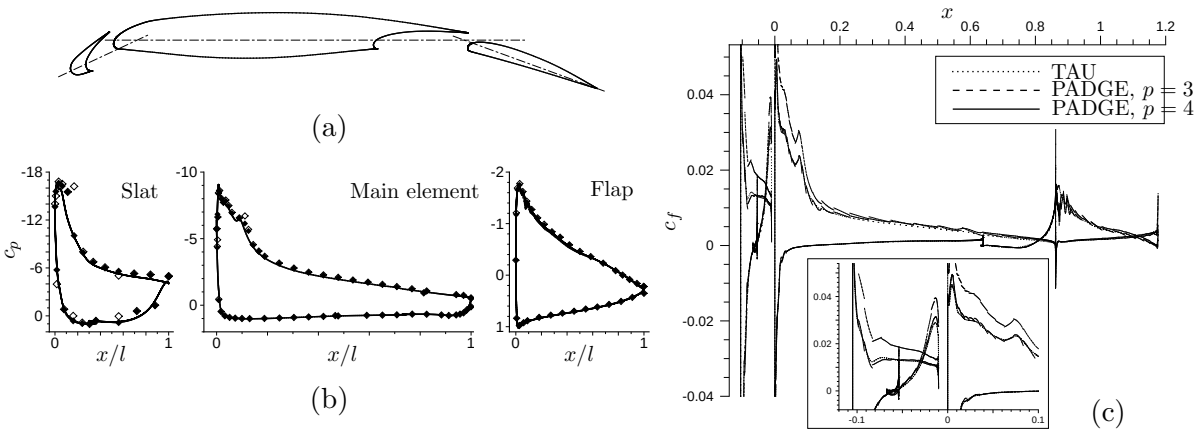


Figure 73: a) Geometry of the L1T2 three-element airfoil. b) Pressure distributions for each L1T2 airfoil element computed by PADGE (solid line) compared to reference results by TAU (dotted) and data of experiment 1 (open symbols) and experiment 2 (filled), c) Comparison of computed skin friction distributions with details of the slat region, [57].

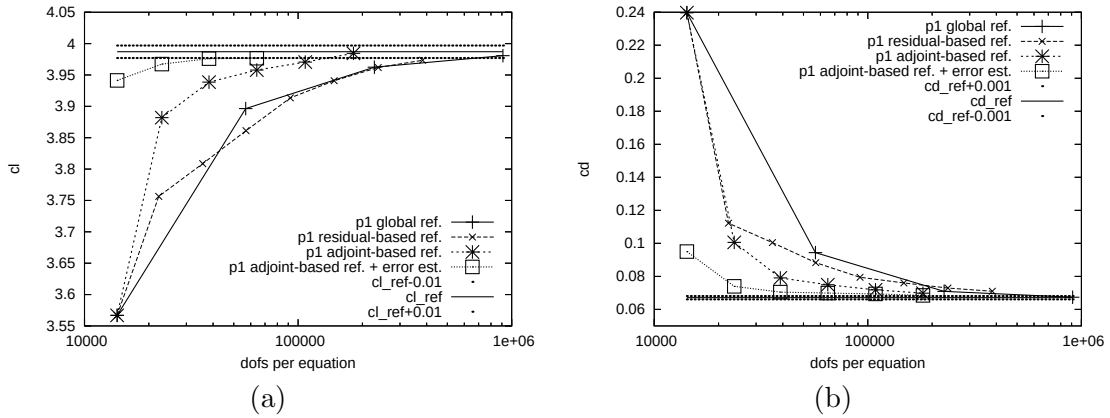


Figure 74: L1T2 high lift configuration: a) lift, $J_{C_l}(\mathbf{u}_h)$, values on globally and on residual-based refined meshes; $J_{C_l}(\mathbf{u}_h)$ and the enhanced values, $J_{C_l}(\mathbf{u}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)$, on adjoint-based refined meshes vs. number of degrees of freedom; b) the respective plot for the drag, $J_{C_d}(\mathbf{u}_h)$, values. Result of the PADGE code [57].

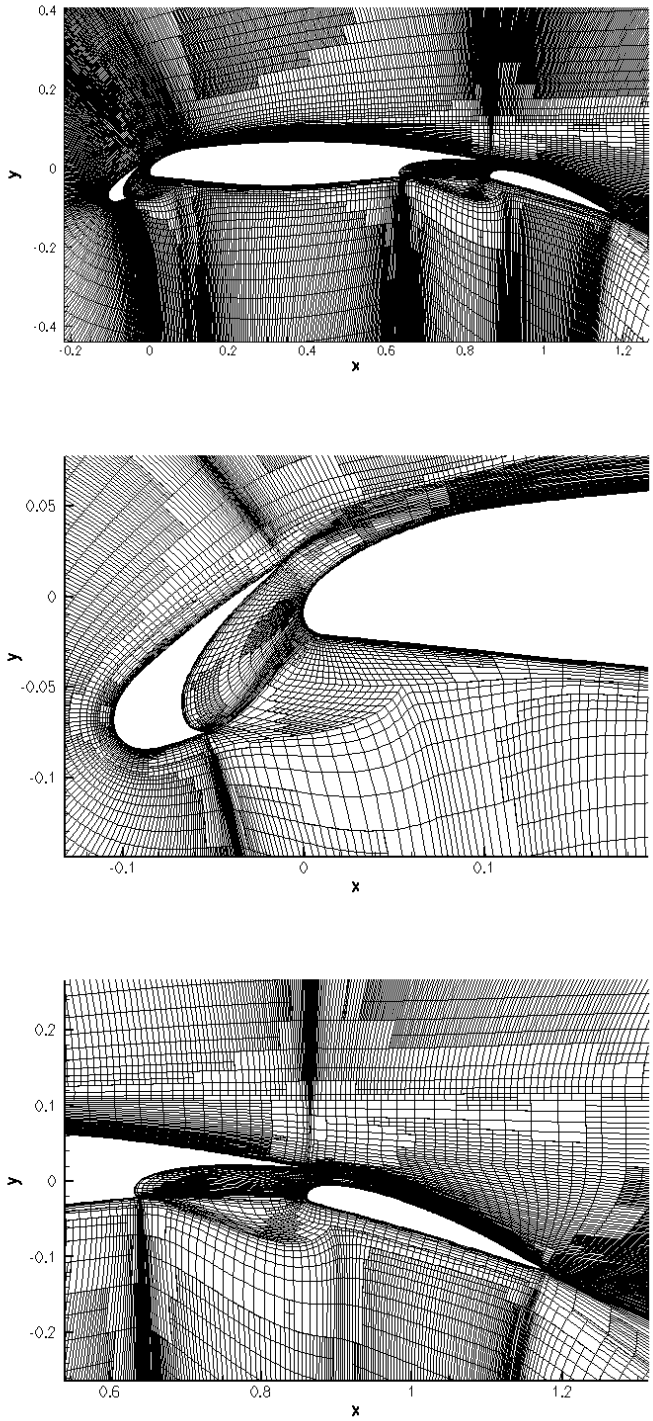


Figure 75: L1T2 high lift configuration: Isotropically residual-based refined mesh of 127536 elements. Result of the PADGE code [57].

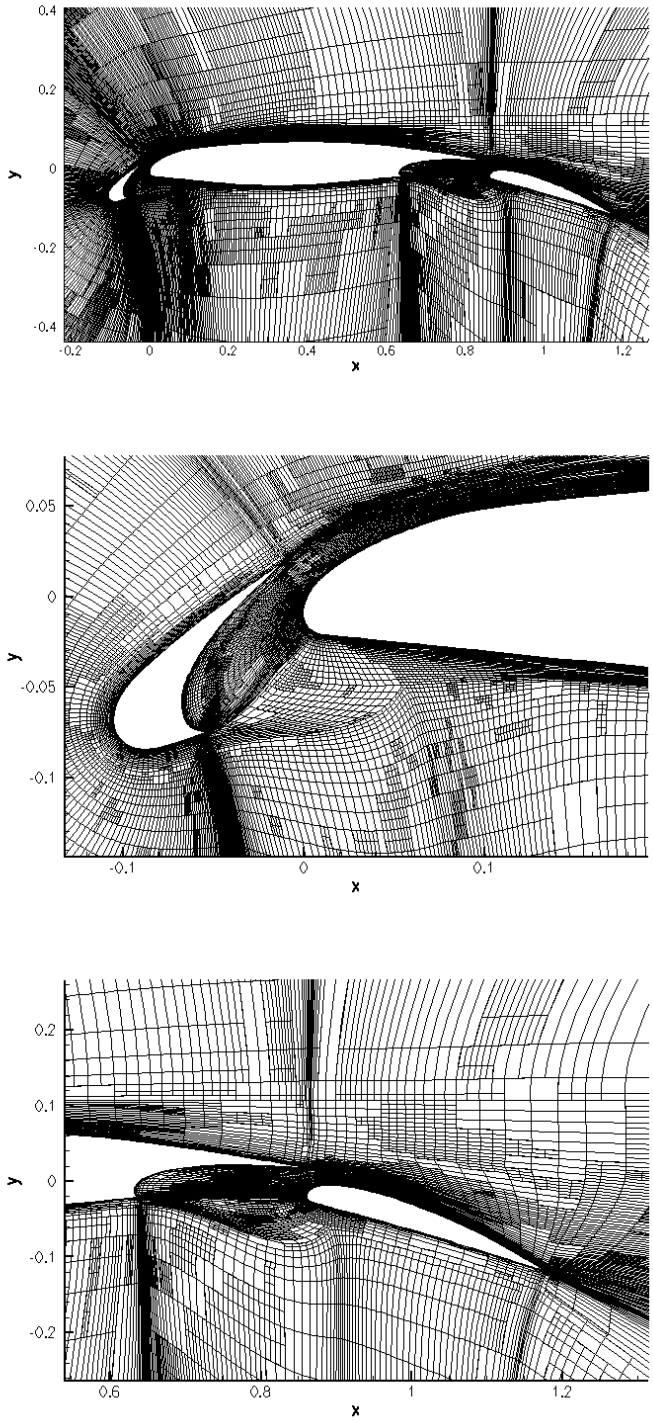


Figure 76: L1T2 high lift configuration: Isotropically adjoint-based refined mesh of 60519 elements for the efficient approximation of C_l. Result of the PADGE code [57].

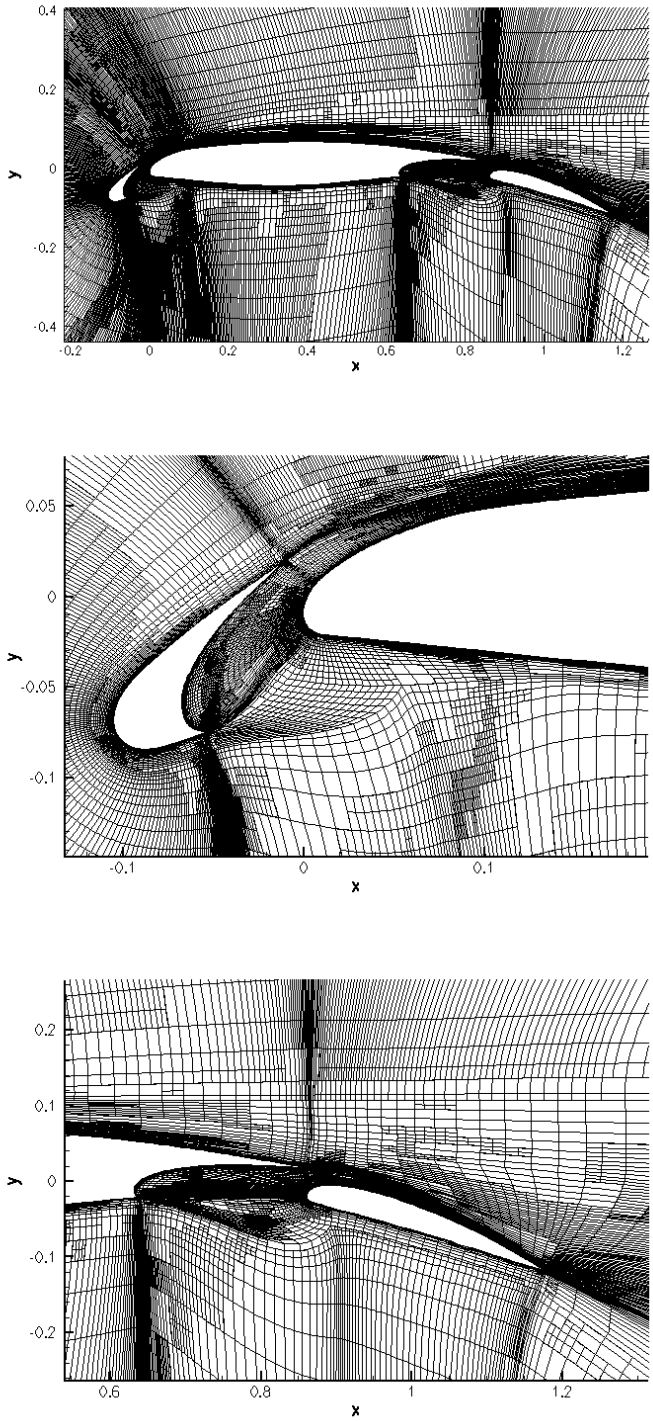


Figure 77: L1T2 high lift configuration: Isotropically adjoint-based refined mesh of 60381 elements for the efficient approximation of C_{d}. Result of the PADGE code [57].

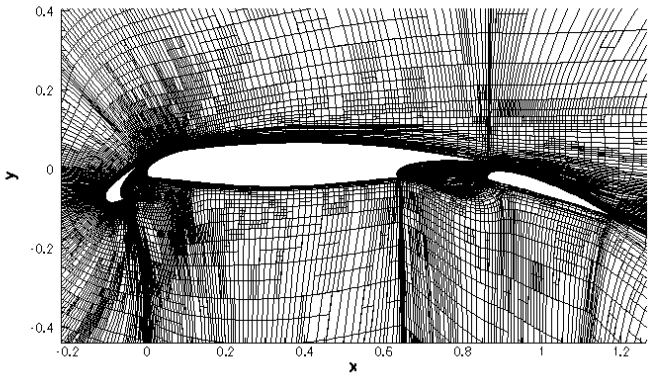


Figure 78: L1T2 high lift configuration: Anisotropically adjoint-based refined mesh of 63085 elements for the efficient approximation of C_{d}. Result of the PADGE code [57].

isotropically refined meshes (including zooms of slat and flap) when using adjoint-based refinement targeted at the lift and drag coefficients C_{l} and C_{d}, respectively. In both cases we see that the mesh has been refined in the neighborhood of the line which separates the recirculation zone behind the slat from the flow which passes between the slat and the main element. We see some refinement in the wake of the slat, main and flap. Furthermore, the shear line emanating from the lower right kink of the main element has been refined. Finally, clearly visible in the middle plot of the C_{d} targeted mesh in Figure 77, the mesh has been refined in the neighborhood of the stagnation line of the main element. We note that, similarly, though not as clearly visible in the given plots, the stagnation lines of the slat and flap are refined. Here, the adjoint solution indicates that the exact position of the stagnation points, as well as the flow upstream of them is particularly important for an accurate prediction of the aerodynamic force coefficients. In comparison to that, in the final residual-based refined mesh shown in Figure 75, we see that more refinement took place in the overall flow field. Also the wakes of slat, main and flap are significantly more refined than in the adjoint-based refined meshes. Finally, we see that no particularly pronounced refinement has been performed at the stagnation lines, which is in clear contrast to the adjoint-based refined meshes in Figures 76 and 77.

We recall, that the meshes in Figures 75, 76 and 77 have been obtain using isotropic refinement. In particular, cell aspect ratios and anisotropies present in the original mesh, see Figure 72, are preserved under isotropic refinement. As can be seen in the figures, while required for more resolution orthogonal to the airfoil geometry isotropic refinement leads to an over-refinement in tangential direction. Also anisotropies introduced in the coarse block-structured mesh which do not match solution anisotropies are preserved through isotropic refinement, although not physically motivated and being of a degrading effect on the solution process and accuracy. In contrast to that, in Figure 78, which shows a *anisotropically* adjoint-based refined mesh for the target C_{d}, we see that due to the anisotropic mesh refinement the elements in a distance to the airfoil have reached a more or less isotropic shape which matches the isotropic behavior of the solution in that region. Also, in the boundary layer anisotropic refinement has been performed which matches the solution behavior better than isotropic mesh refinement.

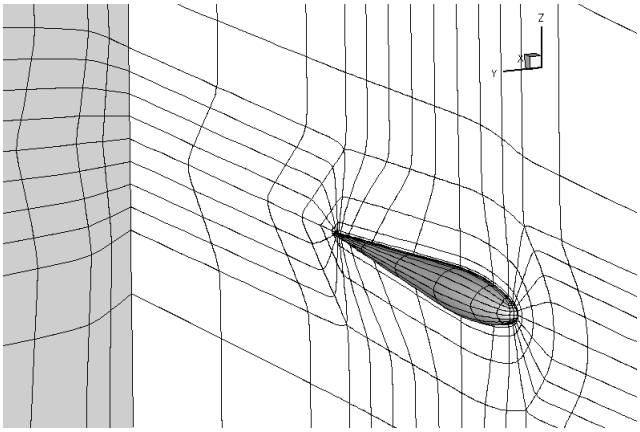


Figure 79: ADGIMA BTC0 test case at turbulent conditions: The coarse mesh with 6656 curved elements. The edges are given by polynomials of degree 4.

7.4 ADIGMA BTC0: Turbulent flow around streamlined body

In this section we consider the ADIGMA BTC0 test case at turbulent flow conditions. In particular, we consider the streamlined body at a Mach number $M = 0.5$, an angle of attack $\alpha = 5^\circ$, and a Reynolds number $\mathrm{Re} = 10 \cdot 10^6$ with adiabatic noslip wall boundary conditions. Reference values $J_{C_\mathrm{l}}(\mathbf{u}) = 0.006612$ and $J_{C_\mathrm{d}}(\mathbf{u}) = 0.0085646$ have been obtained based on higher order computations on very fine grids. The starting mesh of this computation, see Figure 79, has 6656 curved elements. The edges are given by polynomials of degree 4 created by taking additional points from the nested finer grids with straight elements.

First we consider the adjoint-based refinement targeted at efficiently approximating the lift coefficient C_l. In Table 20 we collect the number of elements, the number of degrees of freedom (DoF) of $\mathbf{u}_h \in \mathbf{V}_{h,1}$, the true error $J_{C_\mathrm{l}}(\mathbf{u}) - J_{C_\mathrm{l}}(\mathbf{u}_h)$ in the drag coefficient, the estimated error $\mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h) = \sum_{\kappa \in \mathcal{T}_h} \bar{\eta}_\kappa$, (78), and the effectivity index, i.e., quotient $\theta = \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h) / \left(J_{C_\mathrm{l}}(\mathbf{u}) - J_{C_\mathrm{l}}(\mathbf{u}_h)\right)$ of the estimated and the true error. First of all, we see that on all meshes the sign of the error is predicted correctly. Furthermore, we see that the estimated error is remarkably close to the true error which can also be seen from the quotient θ being close to one.

In Figure 21 we collect the respective data for the adjoint-based refinement targeted at the drag coefficient C_d. Here, we see a similar or even slightly increased accuracy of the error estimation as compared to the lift coefficient in Table 20.

In Figure 80(a) we compare the convergence of C_l for the global, the residual-based and the adjoint-based mesh refinement. We see that within the first refinement step the C_l value for the adjoint-based refinement converges as fast as for the residual-based refinement but both significantly faster than global mesh refinement. However, from the second refinement step onwards the C_l values for the adjoint-based mesh refinement are significantly more accurate than for both residual-based and global mesh refinement.

Furthermore, we see that the error estimation on the adjoint-based refined meshes, already shown in Table 20, further improves the C_l value. In fact, computing the flow solution and its adjoint on the coarsest mesh results in an enhanced C_l value, $\tilde{J}_{C_\mathrm{l}}(\mathbf{u}) = J_{C_\mathrm{l}}(\mathbf{u}) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)$,

# Elements	# DoF	$J_{C_\mathrm{l}}(\mathbf{u}) - J_{C_\mathrm{l}}(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa$	θ
6656	186368	-1.321e-02	-1.310e-02	0.99
16778	469784	-2.096e-03	-2.104e-03	1.00
42699	1195572	-3.572e-04	-3.210e-04	0.90
108940	3050320	-1.686e-04	-1.830e-04	1.08

Table 20: ADGIMA BTC0 case at turbulent conditions: Error estimation for the C_l value.

# Elements	# DoF	$J_{C_\mathrm{d}}(\mathbf{u}) - J_{C_\mathrm{d}}(\mathbf{u}_h)$	$\sum_{\kappa\in\mathcal{T}_h} \bar{\eta}_\kappa$	θ
6656	186368	-1.148e-02	-1.080e-02	0.94
16631	465668	-1.943e-03	-1.924e-03	0.99
41320	1156960	-4.497e-04	-4.263e-04	0.95
102087	2858436	-2.022e-04	-2.022e-04	1.00

Table 21: ADGIMA BTC0 case at turbulent conditions: Error estimation for the C_d value.

which almost coincides with the reference value. Figure 80(b) shows the corresponding error plot. Here we see that the enhanced C_l value already on the coarsest mesh is more accurate than the prescribed ADIGMA tolerance $\texttt{TOL}_{C_\mathrm{l}} = 0.001$ and is even more accurate than the C_l value on the finest adjoint-based, residual-based and globally refined mesh. Also, we see that for a stricter convergence criterion, there is an increasing gain from using adjoint-based refinement in comparison to residual-based and global mesh refinement.

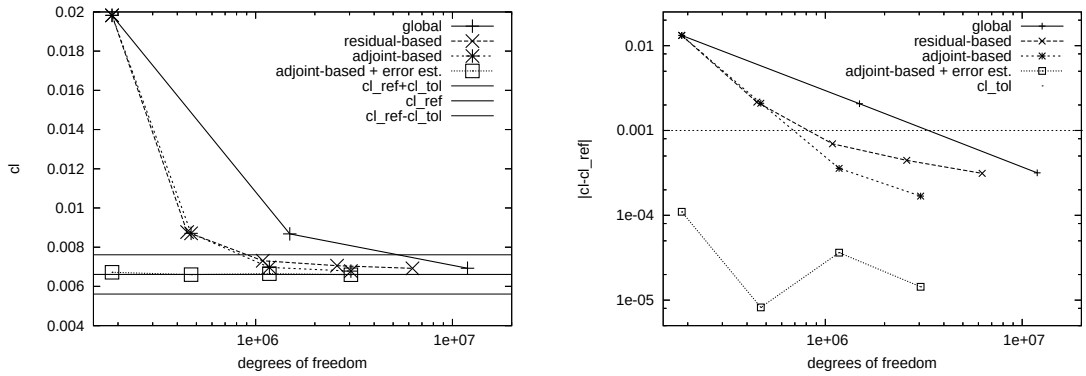
Figure 81(a) and (b) we see the corresponding plots for the C_d value. Here, we see that the enhanced C_d value meets the ADIGMA criterion, $\texttt{TOL}_{C_\mathrm{d}} = 0.0003$, already on the coarsest mesh.

Finally, in Figure 82 we show the final adapted meshes for the adjoint-based and the residual-based mesh refinement. Here, we see that the adjoint-based refinement is mainly concentrated near the airfoil; indeed, the wake is almost unresolved. This corresponds to the fact, that the flow solution in and near the boundary layer is significantly more important for obtaining accurate aerodynamic force coefficients than the flow solution in the wake. In contrast to that the residual-based indicators which are targeted at resolving all flow features also refines elements in the vicinity of the wake.

7.5 ADIGMA CTC4 (modified): Subsonic turbulent flow around DLR-F6 wing-body configuration without fairing

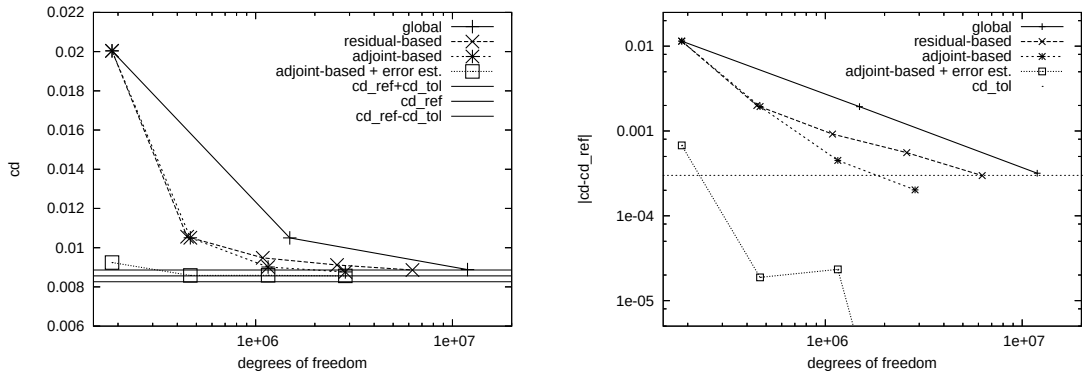
In this final example we consider a turbulent flow at Mach number $M = 0.5$, a Reynolds number $Re = 5 \cdot 10^6$ at an angle of attack $\alpha = -0.141$ around the DLR-F6 wing-body configuration without fairing. This is a modification of the DPW III test case, where a fixed angle of attack has been assumed instead of a given target lift. Also, the Mach number has been reduced from originally $M = 0.75$ to $M = 0.5$ in order to ensure the flow is subsonic.

The original DPW mesh of 3239552 hexahedral elements has been agglomerated twice resulting in a coarse mesh of 50618 hexahedral elements. The additional points of the original mesh have been used to define 50618 curved elements where the curved lines are represented by quartic polynomials. After some regularization this fifth order mesh has been used in a residual-based and an adjoint-based mesh refinement algorithm.



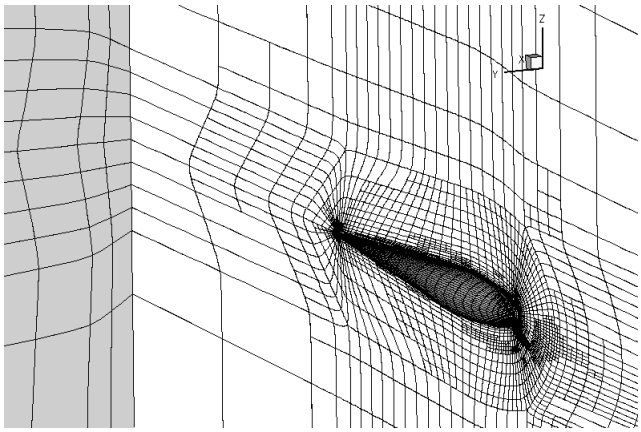
(a) (b)

Figure 80: ADGIMA BTC0 test case at turbulent conditions: (a) lift, $J_{C_\mathrm{l}}(\mathbf{u}_h)$, values on globally and residual-based refined meshes; $J_{C_\mathrm{l}}(\mathbf{u}_h)$ and the enhanced values, $J_{C_\mathrm{l}}(\mathbf{u}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)$, on adjoint-based refined meshes vs. number of degrees of freedom; b) the respective error plot. Result of the PADGE code [57].

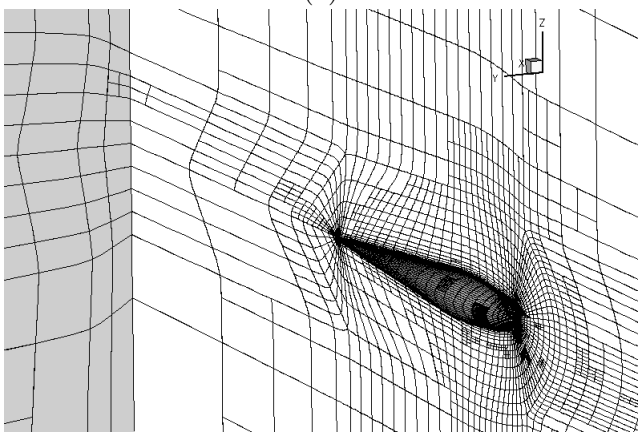


(a) (b)

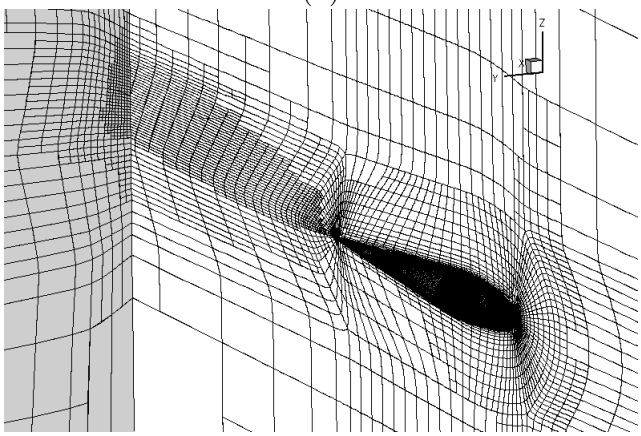
Figure 81: ADGIMA BTC0 test case at turbulent conditions: (a) drag, $J_{C_\mathrm{d}}(\mathbf{u}_h)$, values on globally and residual-based refined meshes; $J_{C_\mathrm{d}}(\mathbf{u}_h)$ and the enhanced values, $J_{C_\mathrm{d}}(\mathbf{u}_h) + \mathcal{R}(\mathbf{u}_h, \bar{\mathbf{z}}_h - \mathbf{z}_h)$, on adjoint-based refined meshes vs. number of degrees of freedom; b) the respective error plot. Result of the PADGE code [57].



(a)



(b)



(c)

Figure 82: ADGIMA BTC0 test case at turbulent conditions: (a) Mesh after 3 adjoint-based refinement steps with target C_l; (b) Mesh after 3 adjoint-based refinement steps with target C_d; (c) Mesh after 4 residual-based refinement steps. Result of the PADGE code [57].

In Figure 83 we see the surface mesh and the c_p distribution on the wing on the coarse mesh of 50618 curved elements. In Figure 84 we see the respective plots for a mesh of 582350 curved elements after 4 residual-based mesh refinement steps. The c_p distribution on the whole wing-body configuration and the mesh on the symmetry plane of the latter mesh are shown in Figure 85. Finally, Figure 86 shows an example of an adjoint-based refined mesh; here for the target quantity C_l, together with the adjoint solution connected to the C_l value.

Acknowledgements

The authors are grateful to Joachim Held, Tobias Leicht, and Florian Prill, as well as to Edward Hall, Manolis Georgoulis, and Stefano Giani for their contributions to the subject of these lecture notes. We note that the PADGE code [57] is based on a DLR modified version of the deal.II library [11, 10]. The first author gratefully acknowledges the partial financial support of the President's Initiative and Networking Fund of the Helmholtz Association of German Research Centres. Both authors acknowledge the partial financial support of the European project ADIGMA [82].

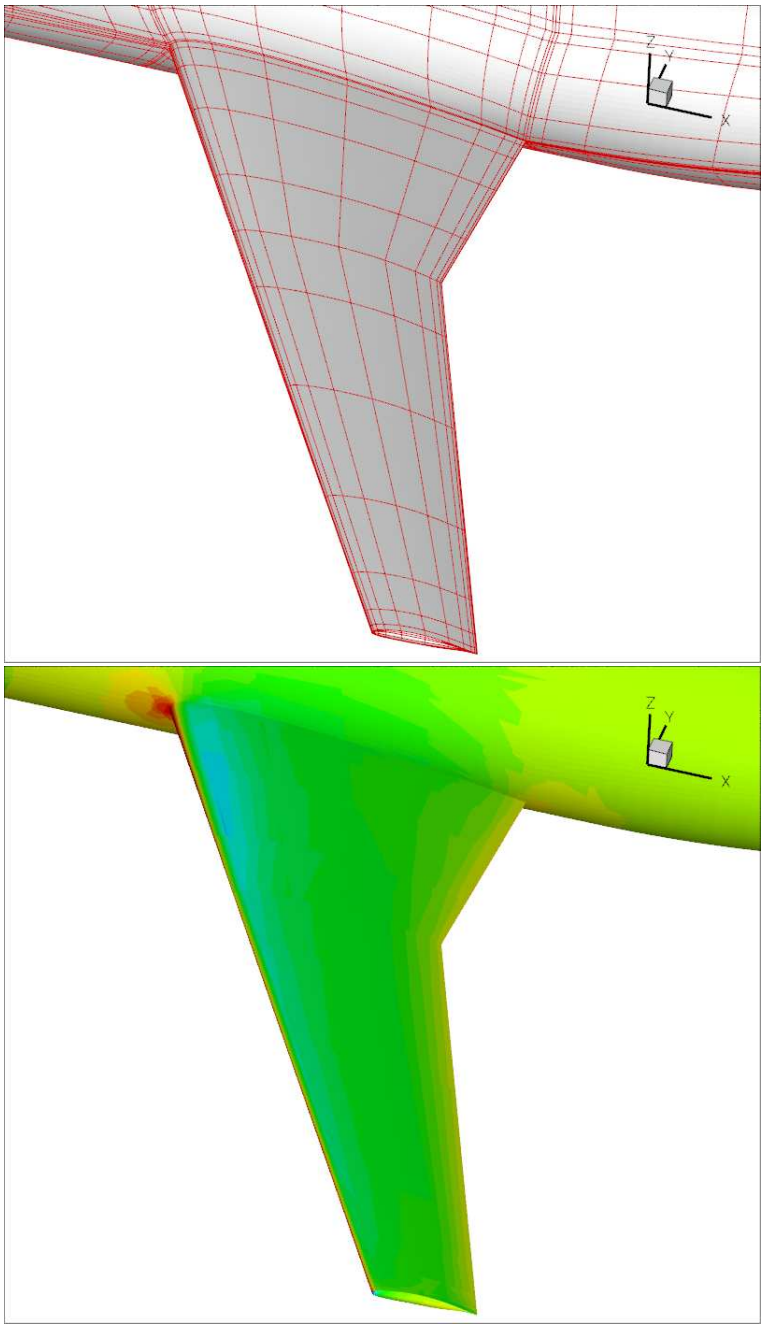


Figure 83: DLR-F6 wing-body configuration: Coarse mesh of 50618 curved elements. a) Surface mesh of wing; b) c_p distribution on wing. Result of the PADGE code [57].

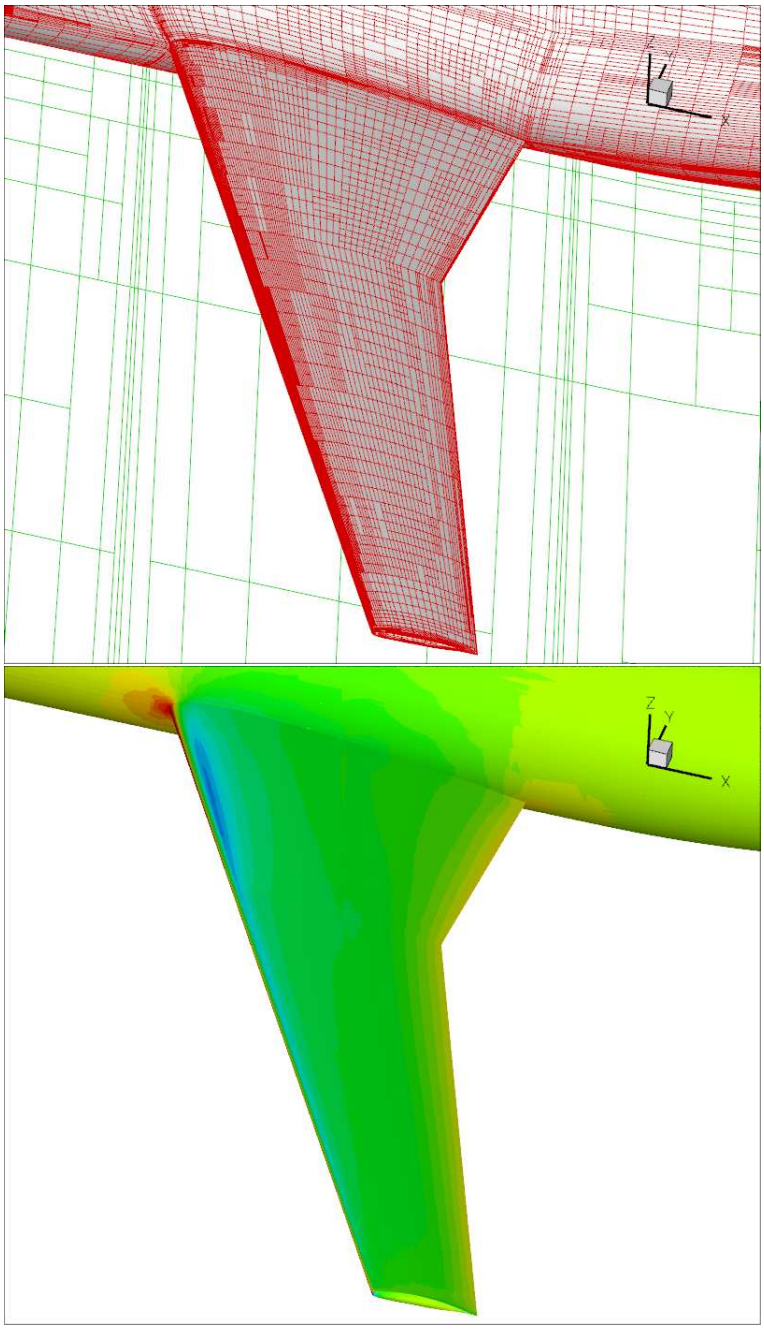


Figure 84: DLR-F6 wing-body configuration: Mesh of 582350 curved elements after 4 residual-based mesh refinement steps. a) Surface mesh of wing; b) c_p distribution on wing. Result of the PADGE code [57].

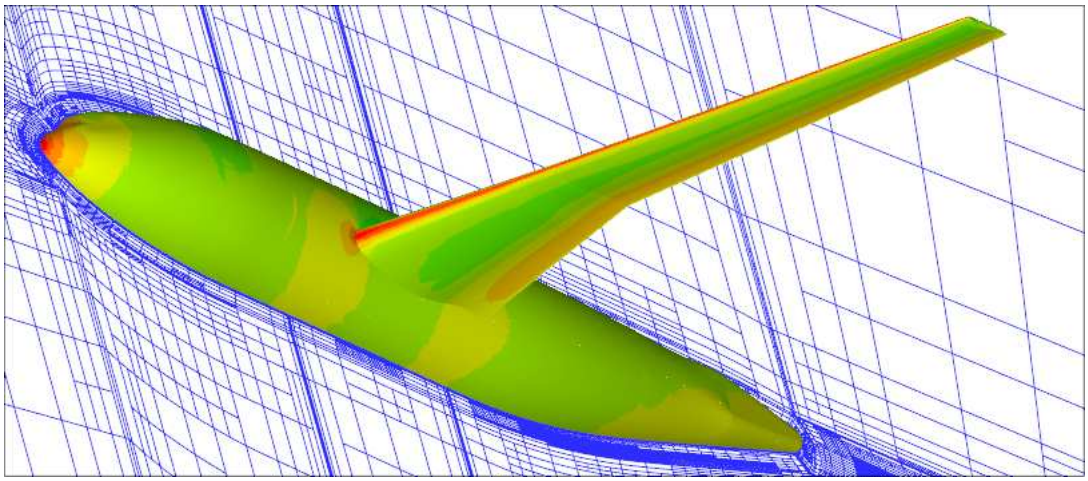
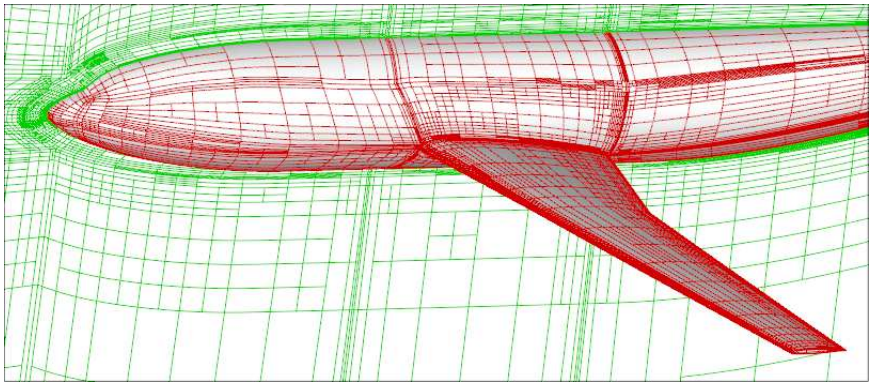
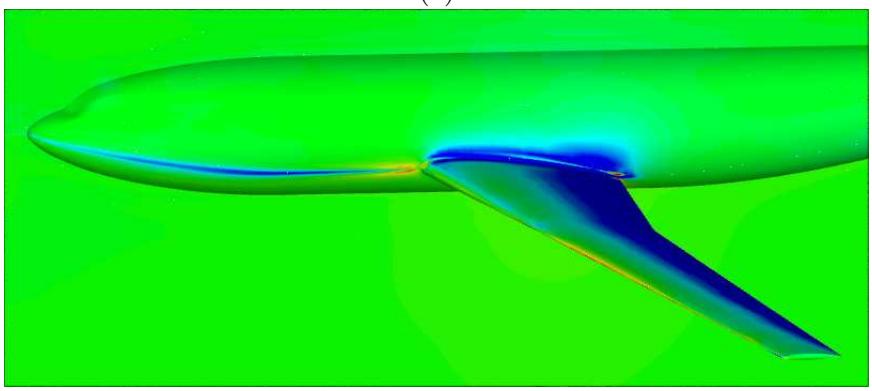


Figure 85: DLR-F6 wing-body configuration: c_p distribution on mesh of 582350 curved elements after 4 residual-based mesh refinement steps. Result of the PADGE code [57].



(a)



(b)

Figure 86: DLR-F6 wing-body configuration: Mesh of 202314 curved elements after two adjoint-based mesh refinement steps targeted at C_l. a) Surface mesh; b) density adjoint distribution, i.e., first comp. of discrete adjoint solution $\bar{\mathbf{z}}_h$. Result of the PADGE code [57].

References

[1] R. Adams. *Sobolev spaces*. Academic Press, New York, 1975.

[2] S. Adjerid, M. Aiffa, and J. E. Flaherty. Computational methods for singularly perturbed systems. In *Analyzing multiscale phenomena using singular perturbation methods (Baltimore, MD, 1998)*, volume 56 of *Proc. Sympos. Appl. Math.*, pages 47–83. Amer. Math. Soc., Providence, RI, 1999.

[3] A selection of experimental test cases for the validation of CFD codes, 1994.

[4] M. Ainsworth and B. Senior. An adaptive refinement strategy for *hp*–finite element computations. *Appl. Numer. Math.*, 26:165–178, 1998.

[5] J. D. Anderson, editor. *Fundamentals of Aerodynamics*. McGraw-Hill, 3rd edition, 2001.

[6] T. Apel. *Anisotropic finite elements: Local estimates and applications*. Advances in Numerical Mathematics, Teubner, Stuttgart, 1999.

[7] D. Arnold, F. Brezzi, B. Cockburn, and L. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002.

[8] I. Babuška and M. Suri. The *hp*–version of the finite element method with quasiuniform meshes. *M^2AN Mathematical Modeling and Numerical Analysis*, 21:199–238, 1987.

[9] G. Baker, W. Jureidini, and O. Karakashian. Piecewise solenoidal vector fields and the Stokes problem. *SIAM J. Numer. Anal.*, 27(6):1466–1485, 1990.

[10] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – A general purpose object oriented finite element library. *ACM Transactions on Mathematical Software*, 33(4), Aug. 2007.

[11] W. Bangerth, R. Hartmann, and G. Kanschat. `deal.II` *Differential Equations Analysis Library, Technical Reference*. http://www.dealii.org/, 6.2 edition, 2009. First edition 1999.

[12] R. F. Bass. *Diffusion and Elliptic Operators*. Spinger–Verlag, New York, 1997.

[13] F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and $k-\omega$ turbulence model equations. *Computers & Fluids*, 34:507–540, 2005.

[14] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comp. Phys.*, 131:267–279, 1997.

[15] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2d Euler equations. *J. Comp. Phys.*, 138:251–285, 1997.

[16] F. Bassi and S. Rebay. GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations. In B. Cockburn, G. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Methods*, volume 11 of *Lecture Notes in Comput. Sci. Engrg.*, pages 197–208. Springer, 2000.

[17] F. Bassi and S. Rebay. Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 40:197–207, 2002.

[18] C. Baumann. *An hp–adaptive discontinuous Galerkin FEM for computational fluid dynamics*. PhD thesis, TICAM, UT Austin, Texas, 1997.

[19] C. Baumann and J. Oden. A discontinuous *hp* finite element method for the Euler and Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 31:79–95, 1999.

[20] C. Baumann and J. Oden. An adaptive-order discontinuous Galerkin method for the solution of the Euler equations of gas dynamics. *International Journal for Numerical Methods in Engineering*, 47:61–73, 2000.

[21] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East–West J. Numer. Math.*, 4:237–264, 1996.

[22] R. Becker and R. Rannacher. Weighted a posteriori error control in FE methods. Technical report, Universität Heidelberg, Heidelberg, Germany, 1996. Preprint 1, Interdisziplinäres Zentrum für Wissenschaftliches Rechnen.

[23] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2001.

[24] C. Bernardi, N. Fiétier, and R. G. Owens. An error indicator for mortar element solutions to the Stokes problem. *IMA J. Numer. Anal.*, 21(4):857–886, 2001.

[25] K. Bey, A. Patra, and J. T. Oden. *hp*-version discontinuous Galerkin methods for hyperbolic conservation laws: a parallel adaptive strategy. *Internat. J. Numer. Methods Engrg.*, 38(22):3889–3908, 1995.

[26] R. Biswas, K. D. Devine, and J. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Appl. Numer. Math.*, 14:255–283, 1994.

[27] W. Cao. On the error of linear interpolation and the orientation, aspect ratio, and internal angles of a triangle. *SIAM J. Numer. Anal.*, 43(1):19–40, 2005.

[28] M. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaption for flow simulations. *Int. J. Numer. Methods Fluids*, 25:475–491, 1997.

[29] G. Chiocchia. Exact solutions to transonic and supersonic flows. Technical Report AR-211, AGARD, 1985.

[30] P. Ciarlet. *The finite element method for elliptic problems*. North-Holland, Amsterdam, 1978.

[31] P. J. Davis. *Interpolation and approximation*. Blaisdell Publishing Co. Ginn and Co. New York-Toronto-London, 1963.

[32] L. Demkowicz, W. Rachowicz, and P. Devloo. A fully automatic *hp*-adaptivity. In *Proceedings of the Fifth International Conference on Spectral and High Order Methods (ICOSAHOM-01) (Uppsala)*, volume 17(1-4), pages 117–142, 2002.

[33] K. D. Devine and J. E. Flaherty. Parallel adaptive *hp*–refinement techniques for conservation laws. *Appl. Numer. Math.*, 20(4):367–386, 1996.

[34] V. Dolejsi. On the discontinuous Galerkin method for the numerical solution of the Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 45:1083–1106, 2004.

[35] T. Eibner and J. M. Melenk. An adaptive strategy for *hp*-FEM based on testing for analyticity. *Comput. Mech.*, 39(5):575–595, 2007.

[36] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. In A. Iserles, editor, *Acta Numerica*, pages 105–158. Cambridge University Press, 1995.

[37] I. Fejtek. Summary of code validation results for a multiple element airfoil test case. 28th AIAA fluid dynamics conference, 1997. AIAA Paper 97-1932.

[38] K. J. Fidkowski and D. L. Darmofal. A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations. *J. Comput. Physics*, 225:1653–1672, 2007.

[39] K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal. *p*-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *J. Comp. Phys.*, 207(1):92–113, July 2005.

[40] L. Formaggia and S. Perotto. New anisotropic a priori error estimates. *Numer. Math.*, 89:641–667, 2001.

[41] E. Georgoulis. Discontinuous Galerkin methods on shape-regular and anisotropic meshes. *D.Phil. Thesis*, University of Oxford, 2003.

[42] E. Georgoulis. *hp*–version interior penalty discontinuous Galerkin finite element methods on anisotropic meshes. *Int. J. Numer. Anal. Model.*, 3:52–79, 2006.

[43] E. Georgoulis, E. Hall, and P. Houston. Discontinuous Galerkin methods for advection–diffusion–reaction problems on anisotropically refined meshes. *SIAM J. Sci. Comput.*, 30(1):246–271, 2007.

[44] E. Georgoulis, E. Hall, and P. Houston. Discontinuous Galerkin methods on *hp*–anisotropic meshes I: A priori error analysis. *Int. J. Comp. Sci. Math.*, 1(2-3):221–244, 2007.

[45] E. Georgoulis, E. Hall, and P. Houston. Discontinuous Galerkin methods on *hp*–anisotropic meshes II: A posteriori error analysis and adaptivity. *Appl. Numer. Math.*, 59(9):2179–2194, 2009.

[46] E. Georgoulis and A. Lasis. A note on the design of *hp*–version interior penalty discontinuous Galerkin finite element methods for degenerate problems. *IMA J. Numer. Anal.*, 26(2):381–390, 2006.

[47] M. Giles and N. Pierce. Adjoint equations in CFD: duality, boundary conditions and solution behaviour. *AIAA*, 97-1850, 1997.

[48] W. Gui and I. Babuška. The h, p and h–p versions of the finite element method in 1 dimension. Part III. The adaptive h–p version. *Numer. Math.*, 49:659–683, 1986.

[49] E. J. C. Hall. *Anisotropic Adaptive Refinement For Discontinuous Galerkin Methods.* PhD thesis, Department of Mathematics, University of Leicester, 2007.

[50] K. Harriman, P. Houston, B. Senior, and E. Süli. *hp*–Version discontinuous Galerkin methods with interior penalty for partial differential equations with nonnegative characteristic form. In C.-W. Shu, T. Tang, and S.-Y. Cheng, editors, *Recent Advances in Scientific Computing and Partial Differential Equations. Contemporary Mathematics Vol. 330*, pages 89–119. AMS, 2003.

[51] R. Hartmann. Adaptive FE Methods for Conservation Equations. In H. Freistühler and G. Warnecke, editors, *Hyperbolic Problems: theory, numerics, applications: eighth international conference in Magdeburg, February, March 2000*, volume 141 of *International series of numerical mathematics*, pages 495–503. Birkhäuser, Basel, 2001.

[52] R. Hartmann. *Adaptive Finite Element Methods for the Compressible Euler Equations.* PhD thesis, University of Heidelberg, 2002.

[53] R. Hartmann. Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 51(9–10):1131–1156, 2006.

[54] R. Hartmann. Adjoint consistency analysis of discontinuous Galerkin discretizations. *SIAM J. Numer. Anal.*, 45(6):2671–2696, 2007.

[55] R. Hartmann. Multitarget error estimation and adaptivity in aerodynamic flow simulations. *SIAM J. Sci. Comput.*, 31(1):708–731, 2008.

[56] R. Hartmann. Numerical analysis of higher order discontinuous Galerkin finite element methods. In H. Deconinck, editor, *VKI LS 2008-08: CFD - ADIGMA course on very high order discretization methods, Oct. 13-17, 2008*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2008.

[57] R. Hartmann, J. Held, T. Leicht, and F. Prill. Discontinuous Galerkin methods for computational aerodynamics – 3D adaptive flow simulation with the DLR PADGE code. *Aerospace Science and Technology*, 2009. Submitted.

[58] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 24:979–1004, 2002.

[59] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.*, 183(2):508–532, 2002.

[60] R. Hartmann and P. Houston. Goal-oriented a posteriori error estimation for multiple target functionals. In T. Y. Hou and E. Tadmor, editors, *Hyperbolic problems: theory, numerics, applications*, pages 579–588. Springer, 2003.

[61] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations I: Method formulation. *Int. J. Num. Anal. Model.*, 3(1):1–20, 2006.

[62] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: Goal–oriented a posteriori error estimation. *Int. J. Num. Anal. Model.*, 3(2):141–162, 2006.

[63] R. Hartmann and P. Houston. An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier–Stokes equations. *J. Comput. Phys.*, 227(22):9670–9685, 2008.

[64] R. Hartmann and T. Leicht. Error estimation and anisotropic mesh refinement for 3d aerodynamic flow simulations. *J. Comput. Phys.*, 2009. Submitted.

[65] A. Hellsten. New Two-Equation Turbulence Model for Aerodynamics Applications. Technical Report Report No. A-21, Helsinki University of Technology, Laboratory of Aerodynamics, 2004.

[66] N. Heuer, M. E. Mellado, and E. P. Stephan. *hp*–Adaptive two–level methods for boundary integral equations on curves. *Computing*, 67(4):305–335, 2001.

[67] V. Heuveline and R. Rannacher. Duality-based adaptivity in the *hp*-finite element method. *J. Numer. Math.*, 1(2):95–113, 2003.

[68] L. Hörmander. *The Analysis of Linear Partial Differential Operators I: Distributional Theory and Fourier Analysis*. Springer–Verlag, 1990.

[69] P. Houston, E. Georgoulis, and E. Hall. Adaptivity and a posteriori error estimation for DG methods on anisotropic meshes. In G. Lube and G. Rapin, editors, *Proceedings of the International Conference on Boundary and Interior Layers (BAIL) - Computational and Asymptotic Methods*. 2006.

[70] P. Houston, J. Mackenzie, E. Süli, and G. Warnecke. A posteriori error analysis for numerical approximations of Friedrichs systems. *Numerische Mathematik*, 82:433–470, 1999.

[71] P. Houston, R. Rannacher, and E. Süli. A posteriori error analysis for stabilised finite element approximations of transport problems. *Comput. Meth. Appl. Mech. Engrg.*, 190(11-12):1483–1508, 2000.

[72] P. Houston, D. Schötzau, and T. P. Wihler. An *hp*-adaptive mixed discontinuous Galerkin FEM for nearly incompressible linear elasticity. *Comput. Methods Appl. Mech. Engrg.*, 195(25-28):3224–3246, 2006.

[73] P. Houston, C. Schwab, and E. Süli. Stabilized *hp*–finite element methods for first–order hyperbolic problems. *SIAM J. Numer. Anal.*, 37:1618–1643, 2000.

[74] P. Houston, C. Schwab, and E. Süli. Discontinuous *hp*-finite element methods for advection–diffusion–reaction problems. *SIAM J. Numer. Anal.*, 39:2133–2163, 2002.

[75] P. Houston, B. Senior, and E. Süli. Sobolev regularity estimation for *hp*–adaptive finite element methods. In F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, editors, *Numerical Mathematics and Advanced Applications ENUMATH 2001*, pages 631–656. Springer, 2003.

[76] P. Houston and E. Süli. Local mesh design for the numerical solution of hyperbolic problems. In M. Baines, editor, *Numerical methods for Fluid Dynamics VI, ICFD*, pages 17–30, 1998.

[77] P. Houston and E. Süli. *hp*–Adaptive discontinuous Galerkin finite element methods for hyperbolic problems. *SIAM J. Sci. Comput.*, 23:1225–1251, 2001.

[78] P. Houston and E. Süli. Stabilized *hp*-finite element approximation of partial differential equations with non-negative characteristic form. *Computing*, 66:99–119, 2001.

[79] P. Houston and E. Süli. Adaptive finite element approximation of hyperbolic problems. In T. Barth and H. Deconinck, editors, *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics. Lect. Notes Comput. Sci. Engrg.*, volume 25, pages 269–344. Springer, 2002.

[80] W. Huang. Mathematical principles of anisotropic mesh adaptation. *Commun. Comput. Phys.*, 1(2):276–310, 2006.

[81] C. M. Klaij, J. J. W. van der Vegt, and H. van der Ven. Space–time discontinuous Galerkin method for the compressible Navier-Stokes equations. *J. Comput. Phys.*, 217(2):589–611, 2006.

[82] N. Kroll. ADGIMA – A European project on the development of adaptive higher-order variational methods for aerospace applications. 47th AIAA Aerospace Sciences Meeting, 2009. AIAA 2009-176.

[83] G. Kunert. *A posteriori error estimation for anisotropic tetrahedral and triangular finite element meshes.* PhD thesis, TU Chemnitz, 1999.

[84] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21:1253–1278, 2000.

[85] T. Leicht. Anisotropic mesh refinement for discontinuous Galerkin methods in aerodynamic flow simulations. Diploma thesis, Dresden University of Technology, 2006.

[86] T. Leicht and R. Hartmann. Anisotropic mesh refinement for discontinuous Galerkin methods in two-dimensional aerodynamic flow simulations. *Int. J. Numer. Meth. Fluids*, 56(11):2111–2138, April 2008.

[87] C. Mavriplis. Adaptive mesh strategies for the spectral element method. *Comput. Methods Appl. Mech. Engrg.*, 116(1-4):77–86, 1994. ICOSAHOM'92 (Montpellier, 1992).

[88] J. Melenk and B. Wohlmuth. On residual-based a posteriori error estimation in *hp*-FEM. *Adv. Comp. Math.*, 15:311–331, 2001.

[89] I. R. M. Moir. Measurements on a two-dimensional aerofoil with high-lift devices. AGARD Advisory Report 303, Advisory Group for Aerospace Research & Development, Neuilly-sur-Seine, 1994. Test case A2.

[90] P. K. Moore. Applications of Lobatto polynomials to an adaptive finite element method: A posteriori error estimates for *hp*–adaptivity and grid-to-grid interpolation. *Numer. Math.*, 94:367–401, 2003.

[91] J. Oden, I. Babuška, and C. Baumann. A discontinuous *hp*-finite element method for diffusion problems. *J. Comput. Phys.*, 146:491–519, 1998.

[92] J. T. Oden and A. Patra. A parallel adaptive strategy for *hp* finite element computations. *Comput. Methods Appl. Mech. Engrg.*, 121(1-4):449–470, 1995.

[93] J. T. Oden, A. Patra, and Y. S. Feng. An *hp*-adaptive strategy. In A. Noor, editor, *Adaptive, Multilevel and Hierarchical Computational Strategies*, pages 23–46. ASME Publications, 1993.

[94] O. Oleinik and E. Radkevič. *Second Order Equations with Nonnegative Characteristic Form*. American Mathematical Society, Providence, R.I., 1973.

[95] S. Prudhomme, F. Pascal, J. Oden, and A. Romkes. Review of *a priori* error estimation for discontinuous Galerkin methods. TICAM Report 00-27, University of Texas, 2000.

[96] W. Rachowicz, L. Demkowicz, and J. Oden. Toward a universal *h–p* adaptive finite element strategy, Part 3. Design of *h–p* meshes. *Comput. Methods Appl. Mech. Engrg.*, 77:181–212, 1989.

[97] H. Schlichting and E. Truckenbrodt. *Aerodynamics of the Airplane*, volume 1. McGraw-Hill, 1979.

[98] R. Schneider and P. Jimack. Toward anisotropic mesh adaptation based upon sensitivity of a posteriori estimates. Technical Report 2005.03, School of Computing, University of Leeds, 2005.

[99] C. Schwab. *p- and hp-FEM – Theory and Application to Solid and Fluid Mechanics*. Oxford University Press, Oxford, 1998.

[100] D. Schwamborn, T. Gerhold, and R. Heinrich. The DLR TAU-code: Recent applications in research and industry. In P. Wesseling, E. Oñate, and J. Périaux, editors, *Proceedings of European Conference on Computational Fluid Dynamics ECCOMAS CDF 2006, Delft The Netherland*, pages 91–100, 2006.

[101] P. Šolín and L. Demkowicz. Goal-oriented *hp*-adaptivity for elliptic problems. *Comput. Methods Appl. Mech. Engrg.*, 193(6-8):449–468, 2004.

[102] E. Süli, P. Houston, and C. Schwab. *hp*–Finite element methods for hyperbolic problems. In J. Whiteman, editor, *The Mathematics of Finite Elements and Applications X*, pages 143–162. Elsevier, 2000.

[103] E. Süli, P. Houston, and B. Senior. *hp*–Discontinuous Galerkin finite element methods for nonlinear hyperbolic problems. *Int. J. Numer. Methods Fluids*, 40(1-2):153–169, 2002.

[104] E. Süli, C. Schwab, and P. Houston. *hp*–DGFEM for partial differential equations with nonnegative characteristic form. In B. Cockburn, G. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications, Lecture Notes in Computational Science and Engineering, Vol. 11*, pages 221–230. Springer, 2000.

[105] L. N. Trefethen and I. D. Bau. *Numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[106] J. Valenciano and R. G. Owens. An *h-p* adaptive spectral element method for Stokes flow. *Appl. Numer. Math.*, 33(1-4):365–371, 2000.

[107] J. van der Vegt and H. van der Ven. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows, I. General formulation. *J. Comp. Phys.*, 182:546–585, 2002.

[108] K. G. van der Zee. An $H^1(P^h)$-coercive discontinuous Galerkin formulation for the Poisson problem: 1-d analysis. Master's thesis, TU Delft, 2004.